The CT/HoTT Percept Filter: Converting Images and Video into Witnessed Transport, Chart Changes, and Coherence

Douglas C. Youvan

doug@youvan.com

www.youvan.ai

December 28, 2025

Most vision systems and most “image processing” pipelines assume an object-first world: detect edges, segment regions, assign labels, and stabilize a scene as a set of persistent things. That approach is powerful, but it produces characteristic failure modes: identity snaps under occlusion, brittle behavior under lighting shifts, and uncanny acceptance or rejection of synthetic media for the wrong reasons. This paper proposes a different deliverable: a perceptual rendering that is relation-first by construction. Given ordinary images or video, the system outputs a structured representation in which identity is carried by transport across time, decompositions are treated as charts valid in specific regimes, and “sameness” is witnessed by explicit consistency evidence rather than asserted by a single label. The goal is not to claim brains implement formal CT or HoTT. The goal is to build a converter that behaves as if it respects compositional structure: multiple parsing routes, explicit chart transitions, boundary-event marking, and coherence scoring as a first-class signal. We present two implementation routes: algorithmic pipelines that build incidence-first cell structures with consistency checks, and learned models trained to output transport witnesses and commutativity-like coherence measures. We also map prior art nearby and state what is new: not merely using higher-structure internally, but rendering it as an audit trail that users can inspect.

Keywords: CT/HoTT percept filter, relation-first vision, witnessed transport, chart change perception, coherence scoring, incidence-first representation, cell complex extraction, sheaf-like consistency, functorial perception stack, video identity transport, boundary events, audit-trail rendering, robust recognition, synthetic media detection.

A collaboration with GPT-5.2-Thinking. 68 pages. CC4.0.

Outline

1. Problem Statement: Why Ordinary Vision Snaps
1.1 Object-first defaults in image processing and model outputs
1.2 The core failure: discrete refactorizations experienced as ontological ruptures
1.3 The proposed fix: change the perceptual primitive, not the scene
1.4 What “conversion” means: a new output format, not a new camera

2. The Target Output: What a CT/HoTT-Style Percept Looks Like
2.1 Percepts as types: families of presentations (charts) rather than templates
2.2 Identity as transport: continuity carried by explicit witnesses
2.3 Soft incidence: graded attachment prior to hard discretization
2.4 Boundary events: marking the transition zone where charts trade stability
2.5 The viewer experience: overlays, meters, and an inspectable audit trail

3. Contribution Claims and Boundaries
3.1 What the paper is not claiming: no “brains compute HoTT” thesis
3.2 What the paper is claiming: a design discipline for robust percept outputs
3.3 Novelty statement: higher-structure as an exported artifact, not a hidden trick
3.4 Success criteria: fewer snaps, better calibration, visible uncertainty

4. Related Work and Prior Art Map
4.1 Consistency-first integration: local-to-global agreement as a signal
4.2 Higher-order data structures: complexes beyond pixels and pairwise graphs
4.3 Equivariance and temporal consistency in video: identity through change
4.4 Category-flavored compositionality: structure preservation as a design rule
4.5 What remains open: a unified “percept rendering” that outputs witnesses, charts, and boundary events

5. Route A: Algorithmic CT/HoTT Filtering Without Training
5.1 From pixels to continuous fields: orientation, curvature, confidence
5.2 From fields to candidate charts: multiple decompositions in parallel
5.3 From charts to incidence: building an incidence graph/category
5.4 Coherence scoring: agreement across routes as a measurable invariant
5.5 Boundary-event detection: locating transition zones rather than snap points
5.6 Output rendering: how to visualize charts, transports, and alarms

6. Route B: Learned CT/HoTT Filtering With Training
6.1 Multi-chart decoders: learning regime-specific renderings
6.2 Transport objectives: predicting transforms and enforcing lift-consistency
6.3 Witness prediction: learning when two parses are equivalent up to re-factorization
6.4 Anti-overtransport mechanisms: refusal and abstention as trained behaviors
6.5 Practical training data: synthetic perturbations, controlled occlusion, lighting sweeps, edit provenance
6.6 What the model exports: not just labels, but witnesses and coherence traces

7. Representation Choices and Their Consequences
7.1 Mesh-first outputs: explicit parts, explicit remeshing, explicit snap risk
7.2 Implicit fields: continuous substrates with late discretization
7.3 Path-first state: identities as transport chains over time
7.4 Hybrid stacks: continuous core, discrete summaries on demand
7.5 How charts are chosen: stability, conditioning, and action constraints

8. The CT/HoTT Filter as a Visual and Video Style
8.1 Chart overlays: showing competing decompositions without confusion
8.2 Transport visualization: flows, arrows, and stability/strain traces
8.3 Boundary visualization: critical zones as ordinary, not alarming
8.4 Coherence alarms: what is flagged and how it is explained
8.5 User-facing controls: scrub time, pin a locus, compare parses

9. Benchmarks and Experiments
9.1 Invariance profiles: occlusion, lighting, and viewpoint sweeps
9.2 Ambiguity handling: non-collapse and continuous reweighting vs flips
9.3 Synthetic media sensitivity: constraint-consistency vs artifact spotting
9.4 The polyhedron-movie task: graded incidence perception and chart overlap
9.5 Metrics: time-to-stability, reversal geometry, coherence AUC, calibrated abstention

10. Applications
10.1 XR training: can humans learn chart-change qualia?
10.2 Scientific imaging: audit trails for segmentation and identity claims
10.3 Accessibility: stabilizing recognition under adverse conditions
10.4 Forensics: contestable signals rather than verdict machines
10.5 Creative tooling: “category picture” aesthetics as an output genre

11. Cognitive Tradeoffs, Limits, and Failure Modes
11.1 Cost profile: compute, latency, and attentional load
11.2 Pathologies: overidentification, undersegmentation, missed local salience
11.3 Coherence laundering: correlated routes that agree for the wrong reasons
11.4 Stop rules: when discretization is necessary for action and language
11.5 Robustness stance: when to refuse transport and say “unknown”

12. Ethics and Governance
12.1 Enhanced recognition and privacy risk
12.2 The temptation of “oracle sight”: coherence is not truth
12.3 Auditability, contestability, and calibrated uncertainty
12.4 Deployment constitution: logging, access limits, red-team gates, refusal modes

13. Conclusion and Forward Work
13.1 Thesis restated: phase changes become chart changes under witnessed transport
13.2 What would count as a genuine advance: exported witnesses that users can inspect
13.3 Next steps: prototype filters, benchmark releases, and responsible rollout

References

Art

1. Problem Statement: Why Ordinary Vision Snaps

The motivating claim of this paper is not that ordinary human vision is "bad," or that modern computer vision is "wrong." The claim is narrower and more diagnostic: both human perception and most contemporary image-processing pipelines are biased toward early discretization. They stabilize the world by committing to objects, parts, and boundaries as soon as possible, because those commitments support action, language, and fast inference. The cost of that bias is a characteristic family of failure modes. When the world (or the data) forces a re-description—when the "best decomposition" changes—our systems often register the event not as a normal transition in representation, but as a rupture in reality. We experience a snap.

This section defines "snap" precisely enough to be testable, ties it to specific defaults in image processing and model outputs, and then states the paper's proposed remedy: shift the primitive from object-first segmentation to relation-first coherence, so that discrete refactorizations become chart changes rather than ontological breaks. Finally, it clarifies what "conversion" means here: not building a new sensor, but producing a new kind of perceptual artifact from ordinary images and video.

1.1 Object-first defaults in image processing and model outputs

The default ontology of vision engineering is object-first. Even when pipelines differ in architecture, they typically share a commitment: the goal of perception is to carve the visual field into stable entities with stable names.

At the classical end, the object-first bias appears as a sequence of commitments:

- compute edges and gradients;
- close contours;
- segment regions;
- fit primitives (lines, corners, planes);
- build object hypotheses;
- attach labels.

Even "soft" methods often harden early. Probabilistic segmentation still aims to output discrete regions. Tracking still aims to output object trajectories. Structure-from-motion still aims to output a discrete 3D reconstruction. The world is made manageable by being discretized.

Modern deep learning often looks different on paper, but the output contracts remain similar:

- detection outputs bounding boxes and class names;
- segmentation outputs pixelwise class maps;
- pose estimation outputs landmark sets;
- re-identification outputs identity labels or embeddings summarized by nearest neighbors.

Even when the internal representation is continuous (a latent vector), the system is trained to stabilize a discrete decision surface: which class, which instance, which mask. Many systems do contain “intermediate” features that could support a more relational rendering, but those features are typically not exported as a perceptual artifact. They are consumed internally and then collapsed into a label or a mask.

This matters because the snap we care about is often created at the collapse boundary: the moment the system commits to one discrete parse as the world. Commitment is necessary for action, but it is not free. It transforms representational uncertainty into ontological certainty. Once that transformation is made, later revisions feel like reality changing rather than descriptions changing.

So the object-first default is not merely a habit. It is a design choice: privileging early discrete commitments because they are convenient, communicable, and actionable. The paper’s claim is that this design choice is the primary source of snap-like failure modes in both humans and machines.

1.2 The core failure: discrete refactorizations experienced as ontological ruptures

A “refactorization” in this paper means a change in the internal decomposition used to describe a scene element—without implying that the scene element itself has ceased to exist. In geometry, a refactorization might be a change in how a surface is partitioned into faces, edges, and vertices as a parameter varies. In video, it might be a change in how the system assigns a set of pixels to one tracked identity rather than another. In segmentation, it might be a topological change: one region splits into two, or two merge into one.

These events are common and, in principle, ordinary. The world is full of them:

- an object rotates and a previously hidden face becomes visible;
- a hand passes in front of a face and occludes key features;
- a person turns sideways and the “best face chart” changes;

- shadows move and edges that looked like boundaries disappear;
- a camera cut forces a new viewpoint regime;
- compression artifacts or edits perturb local cues.

An object-first system tends to interpret these as: “the object changed,” “the identity changed,” or “the segmentation changed.” But the deeper problem is not that the segmentation changed. The problem is that the system treats its segmentation as the object. When the segmentation must be revised, it feels like the object has ruptured.

This is the snap: a representational necessity misread as an ontological event.

In human phenomenology, snaps appear as the sense that a scene “flipped,” that something “became a different thing,” or that the world “re-labeled itself.” In machine behavior, snaps appear as abrupt label flips, brittle tracking breaks, sudden segmentation topology changes, and high-confidence misclassifications that reverse suddenly when the system crosses a decision boundary.

The most important detail is that snaps often happen exactly where a more structural view would predict *boundary events*: transition zones where one description becomes ill-conditioned and another becomes stable. In other words, snaps are not random. They are artifacts of the system’s choice of primitives.

The cube–octahedron story was our toy example: a discrete change in the local incidence census (3 faces vs 4 faces meeting at a vertex) is hard to “see” as smooth if one’s primitive is “a vertex is where exactly k faces meet.” But it becomes ordinary if the primitive is “junction structure is a graded participation field whose discrete census is a chart-level summary.” The same logic applies to recognition in video: if identity is “the label assigned now,” then relabeling is rupture; if identity is “transportable structure across time,” then relabeling is a chart change.

So the core failure is not “models are wrong” but “models collapse into the wrong kind of output.” They collapse continuous, multi-route evidence into a single discrete object token too early, and then any necessary revision is experienced as a discontinuity in being.

1.3 The proposed fix: change the perceptual primitive, not the scene

The proposed fix is a design stance: treat coherence and transport as primitives, and treat objects/parts/labels as derived summaries. This does not deny objects. It changes what the system regards as metaphysically basic.

In the CT/HoTT-inspired framing, the primitive is not “this is the object.” The primitive is:

- here are multiple charts (multiple decompositions) that can represent the same underlying percept;
- here is the evidence that these charts cohere (or fail to cohere);
- here is a transport witness carrying identity and local structure across time and across chart transitions;
- here are boundary zones where one chart becomes ill-conditioned and another becomes stable.

Under this stance, refactorizations stop being ruptures because the system never pretends that a particular decomposition is reality itself. It treats decompositions as coordinate choices with domains of validity. A chart can fail without the object failing.

This is also where the distinction between “coherence-first” and “blurred” must be maintained. The fix is not to make everything fuzzy. The fix is to provide explicit witnesses and explicit failure modes. A coherence-first converter should be able to say:

- “these two parses are equivalent up to a transport witness”;
- “these two parses conflict; no witness exists under current constraints”;
- “this region is a boundary zone; expect chart overlap and higher sensitivity”;
- “a discrete decision is required for action, but it is downstream of the richer state.”

In short, the fix is to upgrade the percept from “a single best segmentation/label” to “a structured account of competing decompositions and the evidence relating them.” That is what makes snaps unnecessary. Snaps arise when the system hides its own refactorizations. The fix is to make refactorization an explicit, ordinary output.

1.4 What “conversion” means: a new output format, not a new camera

The word “conversion” is easily misunderstood. We are not proposing a new sensor modality, a new camera, or a new mystical way of seeing the external world. The input remains ordinary: RGB images, video streams, depth when available. The novelty lies entirely in the output contract.

A standard vision system outputs one of a few familiar products: boxes, masks, labels, poses, embeddings. Our proposed converter outputs a *percept artifact* with different primitives:

- a set of chart candidates (multiple decompositions) with explicit domains of validity;

- graded incidence/attachment fields rather than hard adjacency alone;
- transport witnesses linking identities and local structures across frames;
- coherence scores that measure agreement across multiple parsing routes;
- boundary-event annotations where chart changes are expected and interpretable.

This is why we call it a "filter." It is an image/video-to-structure transform, analogous in spirit to edge detection or optical flow—but with a higher-level deliverable: not edges, but witnessed equivalences; not flow alone, but transport that preserves structure; not segmentation alone, but chart families and transition zones.

The payoff is that the output becomes an audit trail. Instead of asking the viewer to trust that the model's hidden state "knows what it's doing," we export the relevant structure: where the system is confident, where it is strained, what alternative decompositions remain plausible, and what kind of witness (if any) justifies an identity claim.

This is also the main bridge to your earlier "incidence category picture" paper. There, the artifact was a page-sized diagram that made duality visible as a structured reversal of incidence. Here, the artifact is a perceptual rendering that makes identity and refactorization visible as chart changes supported (or not supported) by transport witnesses. In both cases, the value is the same: not only better performance, but better intelligibility of why a structure-preserving claim is being made.

That is the paper's starting point: ordinary vision snaps because it mistakes its own discrete summaries for reality. A CT/HoTT percept filter is a proposal to stop making that mistake—by changing what the system outputs as "what it sees."

2. The Target Output: What a CT/HoTT-Style Percept Looks Like

If Section 1 argued that snaps are often an artifact of early discretization, this section answers the practical question: what do we output instead? "CT/HoTT-style percept" is not meant as a claim about formal proof assistants running in a camera. It is a design specification for an exported representation whose primitives resemble the intuitions we borrowed from category theory and homotopy type theory: compositionality, multiple charts, and witnessed continuity.

A good way to read this section is: we are defining a new file format for perception. Not literally a file extension, but a contract between the converter and its user. Instead of producing only a label or mask, the converter produces a structured object whose internal parts can be inspected and whose changes over time are meaningful. The output should make explicit what ordinary

pipelines hide: competing parses, transport chains, and the loci where a re-description is expected.

2.1 Percepts as types: families of presentations (charts) rather than templates

A template view of perception assumes that there is a best representation—"the face," "the object," "the segmentation"—and that variations (lighting, pose, occlusion) are nuisances that must be normalized away. In practice, this drives systems toward brittle canonicalization: find the canonical view, and match against it.

A type view reverses the emphasis. A percept is not a single image or single parse. It is a *family* of valid presentations tied together by allowable transformations. The "thing" is the family; any particular image-level rendering is a chart on that family.

Concretely, for a CT/HoTT-style percept, this means the converter does not output "the segmentation." It outputs:

- a set of *chart candidates* for the same underlying scene element (e.g., texture chart, contour chart, motion chart, depth chart);
- each chart comes with an explicit *domain of validity* ("works well for frontal faces," "works well for backlit profiles," "works well under partial occlusion," etc.);
- each chart comes with an explicit *conditioning score* (how stable this description currently is).

This is not hand-wavy philosophy; it is a practical way to avoid "this chart failed, therefore the object vanished." Under a chart-family output, a chart can fail locally while the percept persists. The system's job becomes chart management rather than template worship.

One way to think about it: the percept is a manifold-like object with multiple coordinate systems. In certain regimes, one coordinate system is clean; in others, it becomes singular, and another becomes preferable. We do not treat the singularity as the object disappearing. We treat it as a coordinate artifact.

That is the "type" stance. The converter's output must therefore carry multiple presentations in parallel, rather than collapsing them into one early.

2.2 Identity as transport: continuity carried by explicit witnesses

Identity is the most dangerous place to hide machinery. Many systems output an identity label ("Person A") or an embedding vector and call it continuity. But the continuity is implicit: it lives in the trained model and in the user's trust.

A CT/HoTT-style percept exports identity differently: identity is not a label; it is a *witnessed transport*.

The output should include explicit objects of the form:

- *transport witness* from state at time t to state at time t+1, and by composition to longer spans;
- *structure preserved by the transport* (e.g., key relational constraints, geometric correspondences, motion-consistent landmarks);
- *transport strain* (how hard it was to maintain coherence; where the witness is weak).

This is the key move: identity becomes something like "a path" (in the HoTT intuition) rather than "a token equality." The witness is not a proof in a formal system, but an exported explanation object: a bundle of correspondences and transformations that shows why the converter believes this is the same entity across frames.

Several practical consequences follow.

First, continuity becomes inspectable. If the system claims a face in frame 10 and a face in frame 40 are the same person, it must provide the transport chain or an equivalent witness object. If the chain is broken, the system can say so explicitly instead of hallucinating a label.

Second, ambiguity becomes representable without collapse. In difficult scenes, the output can maintain multiple candidate transports ("this track could be Person A or Person B"), each with a witness and a score, rather than forcing a single identity.

Third, synthetic manipulation becomes detectable in the right space. Many deepfakes look plausible frame-by-frame but fail in transport: the deformation paths do not preserve the right constraints. A transport-witness system is naturally sensitive to "failure of lift."

So, in output terms, a CT/HoTT percept contains not only objects but the arrows between them—time-indexed morphisms that justify continuity.

2.3 Soft incidence: graded attachment prior to hard discretization

Incidence is where our earlier polyhedra intuition becomes practical. In object-first systems, adjacency and attachment are typically binary: a pixel is inside a mask or not; a landmark belongs to a face or not; a corner is a vertex or not. This binary stance is exactly what creates snaps when membership must change.

A CT/HoTT percept introduces *soft incidence* as an intermediate layer: attachment is graded before it is discretized.

The output should therefore carry:

- *attachment weights* between elements (pixel-to-region, region-to-part, part-to-object, object-to-scene);
- *incidence confidence* that can ramp smoothly as evidence accumulates;
- *alternative incidence hypotheses* when multiple attachments remain plausible.

This is not "fuzziness for its own sake." It is a way to represent the transition zone honestly. A region does not go from 0% to 100% membership in one frame; it becomes gradually more supported by multiple cues until discretization is warranted.

In the cube–octahedron metaphor, this is the difference between "a vertex has exactly three incident faces" and "this junction has a graded participation of candidate faces; the discrete count is a thresholded readout." In video segmentation, it is the difference between "this pixel belongs to Track 7" and "this pixel's attachment is 0.62 to Track 7 and 0.38 to Track 3 given current cues, and that ambiguity is being carried forward."

Soft incidence also enables a more principled discretization step. Instead of hard thresholding everywhere, discretization can be triggered by action demands: when the system must render a single mask for a downstream controller, it selects a section through the graded attachments, with the choice logged as a decision.

So the output contract includes a graded layer explicitly, with clear points where and why hard decisions are made.

2.4 Boundary events: marking the transition zone where charts trade stability

If charts are real and soft incidence is real, then boundaries are inevitable. The mistake in ordinary pipelines is not that boundaries exist. The mistake is hiding them—collapsing them into sudden flips without explanation.

A CT/HoTT percept therefore must represent *boundary events* explicitly. Not as a single timestamp where "the flip happens," but as a zone where multiple descriptions overlap and trade dominance.

In output terms, the converter should annotate:

- *chart conditioning curves* over time (how stable each chart is);
- *overlap zones* where two or more charts are simultaneously viable;
- *critical loci* where small perturbations change which chart is preferred;

- *event markers* when a discrete summary changes (topology change, identity split/merge, incidence census change), accompanied by the evidence trace that led to the change.

This is where the paper's "snap" story becomes operational. A snap is what happens when an overlap zone is hidden and the system reports only the before and after charts. A boundary-aware percept outputs the overlap zone itself as part of the percept.

The result is not just more honest; it is more useful. In many real-world settings, the overlap zone is exactly where errors concentrate and where users most need interpretability. "Why did the tracker switch IDs here?" "Why did the segmentation split?" The answer is: we entered a boundary zone, chart A became ill-conditioned, chart B took over, and here is the trace.

Boundary events are the perceptual equivalent of logging: a record of where the system's representational regime changed and why.

2.5 The viewer experience: overlays, meters, and an inspectable audit trail

If we succeed, the output should be intelligible to a human without requiring them to study category theory. The math is an inspiration, not the UI.

The viewer experience should therefore be designed around a few simple principles:

- *Show competing charts without clutter.* The user should be able to toggle charts (texture, contour, motion, depth) or view them in parallel as translucent overlays. The point is to make "multiple valid descriptions" visible.
- *Show transport, not just tracks.* Identity should appear as a transport ribbon or flow: arrows linking regions across time, with thickness or intensity representing transport strength. When transport strain rises, the visualization should convey that.
- *Show boundary zones as zones.* Instead of a sudden re-label, the UI should highlight the temporal window where chart overlap existed and where the decision was made.
- *Expose coherence alarms explicitly.* When routes disagree, the system should not silently choose one; it should flag the disagreement and show which constraints are violated.
- *Log every forced discretization.* If the system had to pick a single mask or label for an external consumer, that choice should be recorded as a decision event with alternatives and confidence levels. The audit trail should be inspectable after the fact.

A useful mental model is: the CT/HoTT filter is a "percept debugger." It takes the hidden internal uncertainties and refactorizations that ordinary systems bury, and it exports them as a structured artifact. The artifact supports *replay*: scrub time, pin a region, compare chart

interpretations, inspect transport witnesses, and see exactly where and why a boundary event occurred.

In this sense, the target output is both a representation and a discipline. It is a commitment to treat perception as compositional and revisable without pretending revisions are reality breaks. It is also a commitment to make the system's equivalence claims explicit, so a user can contest them. The CT/HoTT percept is therefore not merely a new kind of model output. It is a new kind of honesty about what perception is doing: maintaining continuity by carrying structure forward, and switching descriptions when the old ones become strained.

3. Contribution Claims and Boundaries

This paper lives in a dangerous rhetorical neighborhood. The moment one mentions "category theory," "HoTT," and "perception," it is easy for the reader to slide into two misunderstandings. The first is mystical: "this is an altered state, LSD geometry, or metaphysical poetry." The second is naive literalism: "the authors claim the brain is running a proof assistant." Neither is what we mean. The safest path is to state boundaries plainly, define contributions as engineering deliverables and evaluation signatures, and make novelty claims in terms of exported artifacts rather than philosophical slogans.

This section is therefore the paper's epistemic hygiene: it constrains what we are promising, and it tells the reader how to judge success without buying the metaphysics.

3.1 What the paper is not claiming: no "brains compute HoTT" thesis

We are not claiming that human neural tissue implements Homotopy Type Theory, maintains formal identity types, or manipulates categorical diagrams as symbolic objects. We are not claiming that the visual cortex literally constructs opposite categories or proves commutative diagrams. We are also not claiming that CT/HoTT is the "true explanation" of perception.

Those claims would be both unnecessary and irresponsible. They would overreach the evidence and invite the wrong kind of critique ("show me the neuron that does univalence"). The point of this paper is to use CT/HoTT as a *design lens* and a *language of constraints*—a way to name and export certain forms of structural behavior that we already care about in vision: compositionality, multi-route consistency, transport across time, and regime-dependent descriptions.

We also explicitly avoid claims of metaphysical superiority. "Object-first vision snaps" is not a condemnation of object-first vision. It is a diagnosis of a particular cost of early discretization. Human vision has good reasons for snapping: action and language demand crispness. Therefore this paper does not position relation-first outputs as replacements for ordinary perception. It

positions them as additional outputs—debuggable percept artifacts that can support robustness, auditing, and training.

Finally, we do not claim that CT/HoTT provides ground truth. It provides a disciplined vocabulary for distinguishing four things that are often conflated: (i) the scene, (ii) a chart used to describe the scene, (iii) the evidence that connects charts, and (iv) downstream commitments for action. Our aim is to keep those distinct.

3.2 What the paper is claiming: a design discipline for robust percept outputs

The positive claim is practical: we propose a design discipline for perception stacks and image/video conversion pipelines that makes refactorization ordinary by exporting the structures that justify continuity.

In concrete terms, the discipline is:

- Treat decompositions as *charts*, not as reality.
- Maintain multiple parsing routes and score their *coherence*.
- Represent identity as *transport witnesses* across time and perturbations.
- Carry *graded attachment* (soft incidence) through transition zones.
- Explicitly mark *boundary events* where charts trade stability.
- Only discretize hard (single mask/label) when required for action, and log the decision.

This is what we mean by "CT/HoTT-style percept," and it is claimable without metaphysical inflation. It says: build outputs that behave as if they respect compositional structure, in the same way that robust numerical methods behave as if they respect conditioning and coordinate changes. Nobody says the physical world "computes Jacobians," but we still use Jacobians because they enforce honest behavior under coordinate change. CT/HoTT is playing that role here: a guide to what honest percept outputs should look like under refactorization.

So the paper's contribution is not "a theorem about perception." It is a blueprint for a new output contract and a family of tests that detect whether a system follows that contract.

3.3 Novelty statement: higher-structure as an exported artifact, not a hidden trick

The core novelty claim is a narrow one, and it is important to keep it narrow to keep it true:

Many existing methods already use "higher-structure-like" ideas internally—multi-view consistency, temporal coherence, equivariance, topological priors, graph diffusion, and so on. But most of those methods collapse their internal reasoning into standard outputs: labels, boxes, masks, embeddings. The higher structure is an internal trick.

We propose making the higher structure the *product*.

That means the converter exports:

- chart families and their conditioning scores;
- explicit transport chains (witnesses) that justify identity claims;
- coherence measures across routes that can be inspected;
- explicit overlap zones and boundary-event annotations;
- decision logs for forced discretizations.

This is the difference between "a model that happens to be robust" and "a percept that shows you how it stays coherent." The latter is a kind of auditability that is rarely treated as the primary output of vision systems.

In other words, the novelty is not that we use incidence, transport, or consistency. The novelty is that we organize them as a unified, user-facing percept artifact—a rendered "category picture" of what the system believes and what justifies that belief—so that refactorizations become visible events rather than silent flips.

If this novelty claim is correct, it shifts the conversation from "accuracy benchmarks only" to "epistemic behavior." A system can be accurate and still be dangerous because it hides when it is guessing. Exported higher structure is a proposal to reduce that danger by making the system's equivalence claims explicit and contestable.

3.4 Success criteria: fewer snaps, better calibration, visible uncertainty

Because we are proposing a design discipline and an output contract, success must be defined in behavioral and perceptual terms, not in slogan terms. The converter succeeds if it changes the *geometry of errors* and the *phenomenology of transitions*.

We propose four success criteria.

First, fewer snaps. This can be measured as fewer abrupt flips in identity, segmentation topology, or discrete structural summaries, *given the same data*—or, more precisely, fewer unaccounted flips. A boundary-aware system may still change its discrete summaries, but it should do so through an overlap zone where the alternative chart is visible and where the decision is logged. A "snap" is a flip with no visible transition regime. Success is converting snaps into chart transitions with overlap.

Second, better calibration. A coherence-first percept should not only be right more often; it should *know when it is likely wrong*. Calibration is measured by the alignment between

confidence and correctness, and by the availability of a principled abstain option. Overtransport is a known failure mode of equivalence-first seeing, so a successful system must show improved abstention behavior in ambiguous regimes rather than overconfident hallucinated witnesses.

Third, visible uncertainty. Uncertainty must not be a hidden scalar. It must be structural: multiple competing charts shown in parallel, multiple candidate transports retained, and coherence disagreements exposed. The viewer should be able to point to the reason for uncertainty: "the motion chart and the texture chart disagree here," rather than merely seeing a low confidence number. Success is uncertainty that is inspectable and localized.

Fourth, useful audit trails. The output should support replay and contestation. If a system claims "this is the same person," it should provide the transport witness and show where it is strong or weak. If it changes its mind, it should record what boundary event caused the transition. Success is not only a performance improvement but an accountability improvement: a human reviewer can evaluate the system's reasons rather than merely accepting its outputs.

Put bluntly: the success of a CT/HoTT percept filter is not that it produces a prettier picture. Success is that it produces a picture that contains the reasons—so that when perception must refactor, it does so openly, continuously when possible, discretely when necessary, and always with a trace that distinguishes chart change from reality change.

4. Related Work and Prior Art Map

This paper is not arriving in a vacuum. Several adjacent literatures already aim at what we might call "relation-first behavior": systems that privilege consistency across views, that represent structure beyond pairwise edges, that enforce symmetry under transformations, or that use categorical language to articulate compositional constraints. The difference is usually *where the structure lives*. In most prior work, higher structure is internal scaffolding; the exported deliverable remains a conventional label, mask, track, or embedding. Our proposal is to treat the higher structure as the delivered artifact: a percept rendering that makes charts, witnesses, and boundary events explicit.

This section maps the neighborhood in five layers, from "local-to-global consistency" up through "category-flavored compositionality," and then isolates what remains open.

4.1 Consistency-first integration: local-to-global agreement as a signal

A long tradition in perception treats *agreement* as the central currency. The world is underdetermined from any single cue, so perception is defined by how multiple partial views cohere.

In classical computer vision, this appears as pipelines that fuse local evidence into a global explanation: you collect local measurements (edges, corners, patch matches, motion vectors) and then solve a global consistency problem (for geometry, correspondence, or segmentation). Even when the math differs—energy minimization, graph cuts, belief propagation, bundle adjustment—the conceptual move is the same: local hints are not trusted individually; they are trusted insofar as they can be made mutually consistent.

In sensor fusion, the same idea becomes explicit: heterogeneous sensors provide partial, locally valid observations, and the fusion problem is deciding what global state (if any) can reconcile them. This is very close to the "chart family" idea in our paper: each sensor or cue is a chart; the global state is the glued object; contradictions are not noise but information about model mismatch or blind spots.

In more modern ML practice, "consistency-first" lives in self-supervision and multi-view learning: you perturb data (cropping, color jitter, time shifts), and you train the system so that representations remain consistent under these changes. The training objective implicitly encodes a transport requirement: a change in presentation should not be mistaken for a change in identity. But the transport is usually implicit and compressed into an embedding. The system behaves as if it is transporting identity, but it does not export the witness.

The relevance to this paper is direct: consistency-first traditions already know that "coherence is signal." Our contribution is to turn coherence into a first-class output, not just a loss function or hidden state. We are taking the consistency objective and making it visible, localized, and inspectable.

4.2 Higher-order data structures: complexes beyond pixels and pairwise graphs

Another adjacent family of work attacks the snap problem by upgrading the representational substrate. Standard vision uses pixels (a grid) and sometimes graphs (pairwise relations). But many real structures are not pairwise; they are inherently higher-order: surfaces, junctions, cells, and multi-part couplings.

This is where simplicial complexes, cell complexes, and hypergraphs enter. The motivation is simple: a triangle is not "three edges" as three independent pairwise links; it is a 2D relation that constrains the three edges together. Similarly, a junction in an image is not just "edges meeting"; it is a higher-order constraint on how orientations, curvatures, and region boundaries cohere at a point or along a curve.

Topological and higher-order deep learning tries to build models that can propagate information not only along edges but across faces, cells, cycles, and higher-dimensional relationships. In neuroscience-inspired contexts, higher-order representations have been used to capture

collective structures that pairwise graphs miss. In geometry-grounded learning, they support reasoning over meshes and surfaces in a way that respects the intrinsic organization of the data.

For our purposes, higher-order substrates are valuable for one main reason: they provide a natural home for "incidence-first" structure. If the converter is going to output charts built from cells and attachments, then it needs a language in which attachments are not afterthoughts. Complexes and cellulations make incidence primary.

Still, most higher-order learning work is aimed at prediction on complex-structured data (classification, regression, generative tasks). The output is typically not a percept rendering with explicit chart overlap, explicit witnesses, and boundary-event annotation. The higher-order structure is the input domain or internal representation, not the exported audit trail. That gap matters, because our goal is not only better performance; it is to make the refactorizations visible.

4.3 Equivariance and temporal consistency in video: identity through change

A third adjacent neighborhood targets the same underlying problem from a different angle: if identity must persist through transformations, then the model should be built to respect those transformations by design.

Equivariance is the principle that when you transform the input in a known way (rotate, translate, permute, apply a motion symmetry), the output should transform correspondingly, not unpredictably. In vision, this idea shows up in convolution (translation equivariance), in rotation-equivariant networks, in SE(3)-equivariant models for 3D, and in newer approaches that try to respect symmetries of motion or flow in sequences.

Video introduces additional structure: time itself is a composition operator. A model that is consistent frame-to-frame is implicitly asserting that it can transport perceptual structure along time. Tracking, optical flow, correspondence, and temporal attention are all attempts to operationalize "identity through change." More recently, temporal consistency has become a primary design constraint in video generation, enhancement, and editing: systems that look good per-frame but drift over time are judged as failures.

This literature matters because it already contains the seed of our "witness" concept. Many video systems compute something witness-like internally: correspondences, flows, warps, or latent trajectories. But those witnesses are typically used and discarded. The user sees only the final track ID, the final generated frame, or the final stabilized output.

Our claim is that video is exactly where witnesses should be exported. The time axis makes composition unavoidable: if you have a transport from t to t+1, and another from t+1 to t+2, the composed transport is a meaningful object. A CT/HoTT-style percept wants to render that

compositional structure, display strain where composition becomes unstable, and mark boundary zones where chart changes become necessary.

So the prior art gives us two important lessons: (i) build symmetry and consistency into the model because post-hoc fixes are brittle, and (ii) time provides a privileged arena where "identity as transport" is not philosophy but engineering. Our proposal extends those lessons into an output contract: export the transport chain, not just the transported label.

4.4 Category-flavored compositionality: structure preservation as a design rule

There is also a growing body of work that uses category-theoretic language to articulate compositionality in learning systems: how complex behaviors are built from simpler ones, how modules compose, how representations respect structure-preserving maps, and how architectures can be understood as morphism factories rather than as opaque function approximators.

This tradition is diverse. Some of it is foundational (using categorical semantics to reframe learning pipelines), some is practical (compositional program-like architectures), and some is organizational (treating systems as networks of components with explicit interfaces). The shared theme is that structure-preserving behavior is not a decorative property; it is a method for building systems that generalize under recombination.

For this paper, the value of "category-flavored compositionality" is not that it proves our percept filter is correct. The value is that it legitimizes a design rule: perception systems should behave well under composition and under re-description. That is exactly what we mean by chart changes and witnessed transport. If a perceptual claim cannot be stably transported across transformations, then the system should treat that claim as fragile. If multiple routes to the same claim disagree, that disagreement should be surfaced.

So, category language here plays the role of a discipline: it encourages the system designer to ask "what is being preserved," "what composes," and "where do equivalences live." Our proposal simply insists that those questions should not stop at architecture diagrams; they should appear in what the system outputs.

4.5 What remains open: a unified "percept rendering" that outputs witnesses, charts, and boundary events

Despite all of the above, something is still missing in the ecosystem: a unified converter whose exported product is a perceptual artifact containing three things simultaneously:

- charts (multiple decompositions with explicit domains and conditioning),
- witnesses (explicit transport objects justifying identity and continuity), and

- boundary events (overlap zones where charts trade stability, logged as ordinary transitions rather than hidden flips).

Most existing systems cover one or two of these partially:

- Consistency-first approaches often give you coherence internally but not an exported witness.
- Higher-order representations often give you incidence-friendly internal structure but not chart overlap and boundary rendering as a user-facing artifact.
- Video-equivariant and temporally consistent methods often compute transport implicitly but do not expose it as an inspectable chain with strain and failure modes.
- Category-flavored compositionality often clarifies design principles but does not yield a concrete, visualizable output standard for perception.

What remains open, therefore, is not “do these mathematical ideas exist,” but “can we package them into an output contract that changes how perception is experienced and audited?”

That open space has several sub-problems that our later sections will treat as design questions rather than as missing theory:

- How to represent chart families without overwhelming the viewer.
- How to define witnesses that are strong enough to be meaningful but cheap enough to compute.
- How to avoid overtransport (false equivalence) while still preventing snap-based collapse.
- How to mark boundary zones as zones (with overlap and strain) rather than as single flip moments.
- How to standardize evaluation so that success is measured not only by accuracy, but by the geometry of revisions: continuous reweighting, calibrated abstention, and visible uncertainty.

In short: the pieces exist in adjacent fields, but the *unified percept rendering*—a converter that exports structural reasons, not just decisions—appears to be underexplored as a primary deliverable. This paper’s aim is to define that deliverable sharply enough that future work can implement it either algorithmically or via training, and evaluate it in terms that match the motivating phenomenon: fewer snaps, more witnessed continuity, and a different kind of ordinary.

5. Route A: Algorithmic CT/HoTT Filtering Without Training

Route A is the "no training set required" path. It treats a CT/HoTT-style percept filter as an algorithmic compiler: pixels go in, and a structured percept artifact comes out. The aim is not to beat the best learned models on benchmark accuracy. The aim is to produce the *right kind of exported structure*: multiple charts, explicit coherence signals, and boundary-event markings that turn flips into explainable transitions.

This route is attractive for three reasons. First, it is auditable end-to-end: every output can be traced to an explicit computation rather than to a learned latent. Second, it can be used as a reference implementation and as a generator of weak supervision for Route B. Third, it forces clarity: we must specify what a "chart," "incidence," "witness," and "boundary event" mean operationally.

Below is a concrete blueprint. It is written as a pipeline, but it should be understood as a set of interacting modules that can be run in parallel.

5.1 From pixels to continuous fields: orientation, curvature, confidence

The starting move is to *avoid discretization as long as possible*. Instead of jumping immediately to edges, corners, and segments, we first compute continuous fields that describe local structure with uncertainty.

A minimal field stack includes:

- **Orientation field**: local dominant direction(s) of intensity change or texture flow. In smooth areas this is undefined; in edge-like areas it is sharp.
- **Edge strength / gradient magnitude field**: where boundaries are likely, but still as a real-valued field, not a binary edge map.
- **Curvature field**: how rapidly the orientation changes along candidate contours; high curvature highlights corners and junctions without committing to "this is a vertex."
- **Texture coherence field**: measures of local anisotropy, periodicity, or stationarity that indicate whether a region is "chartable" by a texture model.
- **Motion/temporal fields (for video)**: optical-flow-like fields, but carried with confidence, plus divergence/curl-like diagnostics that highlight inconsistent motion.
- **Confidence fields**: for each field above, a measure of how well-defined it is locally (signal-to-noise, local fit residuals, temporal stability).

The key is that each field is a *candidate chart seed*. Orientation and curvature seed contour charts; texture coherence seeds region charts; motion seeds transport charts. Confidence is

crucial because the CT/HoTT stance is “charts have domains.” If a field is undefined or ill-conditioned in a region, that is not an error; it is a chart boundary signal.

A practical habit here is to represent each field not as a single value but as a small local model with residuals (e.g., local line fit for orientation, local quadratic fit for curvature). Residuals are what later become coherence alarms: when two charts disagree, the disagreement should be measurable as incompatible residual patterns.

5.2 From fields to candidate charts: multiple decompositions in parallel

Instead of producing one segmentation or one edge map, we produce multiple *candidate charts*, each a partial, locally valid decomposition of the image.

Typical chart families:

- **Contour chart**: a set of candidate boundary curves with uncertainty thickness (think “tubes” rather than lines), derived from edge/orientation fields.
- **Region chart**: a soft partition into coherent regions, derived from texture coherence and color homogeneity fields.
- **Junction chart**: candidate junction loci where multiple contour tubes meet or cross, derived from curvature and orientation conflict patterns.
- **Surface chart (if depth or stereo exists)**: local surface patches with normals and confidence, yielding a geometric chart independent of texture.
- **Motion chart (video)**: coherent motion regions and their local transforms, yielding a temporal chart.

Each chart is incomplete by design. It does not claim to explain the entire frame. It claims to explain certain regions well and leaves other regions as “not charted” or “chart ambiguous.”

This is the crucial difference from standard pipelines. Standard pipelines push toward total coverage: every pixel gets a label. CT/HoTT filtering tolerates “holes” because holes are where chart changes and boundary events live.

Operationally, each chart comes with:

- a *coverage mask* (where it is defined and stable),
- a *conditioning score* (how stable its local fits are),
- a *soft assignment structure* (graded membership of pixels/patches to chart elements),

- and a set of *predicted constraints* (e.g., boundary continuity, region interior smoothness, motion smoothness).

At this stage, we already have the beginning of the target output: not a single parse, but a family of partial parses.

5.3 From charts to incidence: building an incidence graph/category

Now we turn charts into an explicit incidence structure. This is where the earlier cube–octahedron incidence-category language becomes concrete in vision terms.

We define a small set of "cell-like" objects extracted from charts, but kept soft:

- **0-cells (points)**: junction candidates, corners, high-curvature points, motion event points.
- **1-cells (curves)**: boundary curve candidates (contour tubes) and motion trajectories.
- **2-cells (regions)**: region candidates from the region chart, plus surface patches when available.
- **(optional) 3-cells (volumes)**: in multi-view or depth settings, volumetric regions.

Incidence relations are then defined as graded attachments:

- point-to-curve attachment: does a junction lie on a boundary curve?
- curve-to-region attachment: is this curve part of this region boundary?
- point-to-region attachment: does this point sit at the corner of this region?
- curve-to-curve attachment: do these curves meet or cross, and with what confidence?

Instead of binary "is incident to," we store weights and residuals. The residual is the evidence: distance-to-curve, boundary alignment mismatch, motion discontinuity mismatch, etc. The weight is the belief strength.

Once we have these objects and graded incidence relations, we can organize them as an incidence graph. If one wants to honor the category-theory language more explicitly, one can treat each cell as an object and each incidence relation as a morphism, with composition corresponding to higher incidence (point→curve→region implies point→region). The key is not to burden the implementation with formalism; the key is to preserve the compositional constraint: if the system claims point→curve and curve→region with high confidence, then it should also support point→region as a derived attachment with compatible confidence. When this fails, it is a coherence alarm.

So the incidence structure is both a representation and a consistency engine. It is where "composable perception" becomes operational.

5.4 Coherence scoring: agreement across routes as a measurable invariant

Now we define coherence as an explicit measurable quantity. Coherence is not "the model feels good." It is agreement among independently derived constraints.

Coherence scoring can be organized into a few families:

- **Diagram agreement**: does point→curve→region composition agree with direct point→region evidence? If not, the discrepancy is a localized coherence defect.
- **Cross-chart consistency**: do region boundaries predicted by texture align with contour tubes predicted by orientation? Do motion boundaries align with appearance boundaries?
- **Temporal coherence**: do transported cells (curves/regions) remain consistent with freshly extracted cells in the next frame?
- **Redundancy checks**: do different cues predict the same attachment with comparable residuals?

In practice, this becomes a set of scalar fields over the image (or over the cell graph):

- a coherence score per cell (how well it fits into multiple constraint loops),
- a coherence score per incidence relation,
- and a coherence score per region of the image indicating "agreement density."

These scores are the heart of the Route A CT/HoTT filter. They supply the "felt invariant" analog: coherence becomes a quantity that can be rendered. They also supply the mechanism for resisting snaps: when one chart begins to fail, the system can shift weight toward another chart not because of a hard threshold, but because coherence gradually rebalances.

Critically, coherence scores also enable principled abstention. When coherence is low across all charts, the system can output "boundary zone / uncertain" rather than forcing a discrete parse.

5.5 Boundary-event detection: locating transition zones rather than snap points

Boundary events are where refactorizations happen: region splits/merges, identity swaps, topology changes, emergence of a new junction, disappearance of a boundary, and so on. Ordinary pipelines produce a snap because they do not represent the transition zone. Route A explicitly tries to detect and label the zone.

Operationally, a boundary zone is characterized by:

- declining conditioning of a currently dominant chart in a local region,
- rising conditioning of an alternative chart,
- increasing disagreement across cues (coherence dips),
- and/or increased sensitivity to small perturbations (instability in assignments).

So boundary-event detection becomes a detection of overlap regions where:

- chart A and chart B both partially cover the area,
- their predictions disagree in structured ways,
- and soft incidence weights are in flux.

In video, boundary zones are often visible as temporal windows where transport witnesses weaken: the transported structure does not align well with new evidence, yet no alternative transport dominates cleanly. This is precisely the moment where ID switches happen in trackers. A CT/HoTT filter should mark this window as "transport strain rising" rather than silently flipping.

Importantly, boundary-event detection is not only for error avoidance; it is itself part of the output. Boundary zones are where an auditor wants attention. The filter should record:

- start/end times (or frame indices) of the boundary zone,
- which charts are competing,
- what constraints are conflicting,
- and what decision (if any) was made and why.

This is the perceptual analog of a compiler warning: "you are crossing a representation boundary; expect refactorization; here is the reason."

5.6 Output rendering: how to visualize charts, transports, and alarms

Route A lives or dies by rendering. If the output is not legible, the paper's "percept artifact" collapses back into hidden internal computation.

A practical rendering package includes:

- **Chart overlays**: toggleable layers for contour tubes, region partitions, junction candidates, motion regions. Each layer should be soft: thickness indicates uncertainty, opacity indicates conditioning.

- **Incidence visualization**: show attachments as weighted links (point-to-curve, curve-to-region) with thickness indicating confidence and color/texture indicating residual strain (without requiring the user to read numbers).
- **Transport ribbons (video)**: render transport witnesses as arrows or ribbons connecting corresponding cells across frames, with a strain meter that rises when alignment residuals rise.
- **Coherence heatmaps**: display local coherence as a heat overlay, highlighting regions where multi-route agreement is strong versus weak.
- **Boundary zone markers**: mark temporal windows and spatial loci where chart competition and coherence dips occur; allow a scrubber to replay the transition and inspect which chart took over.
- **Alarm panel (audit trail)**: list coherence breaks as "events" with the supporting evidence: which constraint loop failed, which charts disagreed, what was the system's abstain/commit decision.

Two UI rules matter.

First, *never hide alternative charts when the decision is fragile*. If chart competition exists, the viewer should see the competition. This is how we prevent snaps: the overlap zone becomes visible.

Second, *separate perception from commitment*. The percept artifact can retain soft incidence and multiple candidates, while a downstream "commit layer" selects a single mask or label if needed. The commit layer's choices must be logged. This makes accountability possible: we can later ask whether the commitment was forced by an application constraint or whether the percept itself was confident.

Route A therefore outputs not just a picture but a navigable explanation object. It is a perception debugger: it shows where the system is confident, where it is strained, what alternatives remain viable, and where boundary events occur. Even if its raw accuracy were lower than a learned model, it could still be valuable as a ground-truthing tool for epistemic behavior—because it externalizes the very structure that snaps usually conceal.

In short, Route A gives us an implementable baseline CT/HoTT filter: a multi-chart, incidence-explicit, coherence-scored pipeline that marks boundary zones and exports witnesses as inspectable artifacts. It is the algorithmic skeleton on which Route B can later add learned flesh, without losing the auditability that makes the whole project worth doing.

6. Route B: Learned CT/HoTT Filtering With Training

Route A gives a clean, auditable skeleton, but it will eventually collide with the complexity of real imagery: hair, motion blur, specularities, clutter, compression artifacts, and the long tail of "ordinary weirdness." Route B accepts that modern vision performance often comes from learned priors. The challenge is to use learning without collapsing back into the object-first failure mode: a single hidden representation that emits a single discrete decision, with all coherence machinery buried.

So Route B is not "train a model and hope it behaves." Route B is *train a model to export the right kind of structure*: chart families, transport witnesses, coherence traces, and boundary zones. Learning is used to make charts sharper and witnesses stronger, not to replace the discipline with an opaque oracle.

A guiding slogan for Route B is: **learn the charts and the witnesses, and train the model to know when it cannot lift.**

6.1 Multi-chart decoders: learning regime-specific renderings

The chart idea becomes concrete in network design as *multiple decoders* that produce different, partially overlapping renderings of the same scene element. These are not "ensembles" in the usual sense. They are intended to be legitimate coordinate systems with different domains of validity.

A useful architecture pattern is:

- a shared encoder producing a rich latent state for each frame (or frame window),
- plus a set of specialized decoders, each trained to be strong in a particular regime.

Examples of chart decoders:

- **Contour decoder**: outputs soft boundary tubes, junction heatmaps, curvature cues.
- **Region decoder**: outputs soft partitions with uncertainty.
- **Keypoint/landmark decoder**: outputs structured point sets with confidence.
- **Depth/surface decoder** (if data supports it): outputs local surface normals/patches.
- **Motion/flow decoder** (video): outputs flow + confidence + motion segmentation.
- **Identity/instance decoder**: outputs instance hypotheses but explicitly as *multiple candidates*, not a single label.

The crucial point is that each decoder must also output its *conditioning*: a confidence that is calibrated to the decoder's regime. "Confidence" here is not a global scalar; it is a statement like: "the contour chart is stable here; the texture chart is unstable here." That makes chart choice itself an output of the system, not an implicit switch hidden in the network.

To prevent the model from learning a trivial "everyone agrees all the time" story, we want controlled disagreement. In training, we should present stimuli where some charts are systematically disadvantaged (e.g., heavy blur hurts contours; strong shadows hurt texture; occlusion hurts landmarks). The model should learn not to force agreement but to learn where each chart is reliable.

In other words, multi-chart decoders embody the paper's metaphysics in a practical way: multiple descriptions coexist; the converter exports them all; and stability is regime-dependent.

6.2 Transport objectives: predicting transforms and enforcing lift-consistency

To make identity "transport," the model must do more than output consistent embeddings. It must output *transformations* and be penalized when those transformations do not preserve structure.

A transport objective can be built in layers.

First, predict a transform between adjacent frames (or short windows). The transform might be:

- dense flow fields (pixel-level transport),
- piecewise rigid/affine transforms (region-level transport),
- articulated transforms (for faces and bodies),
- latent-space transforms (transport in representation space).

Second, require **lift-consistency**: if you transport a chart-rendering from time t to time t+1 using the predicted transform, it should match the chart-rendering directly produced at t+1.

This creates a commutativity-like constraint:

- render chart at t, then transport it forward, should agree with
- render chart at t+1 directly.

This is the practical meaning of "lift succeeds." The model is not allowed to claim continuity unless it can carry its own structure across time.

Third, force transport to respect *multiple charts*. It is easy to satisfy lift-consistency for one decoder and fail for others. A CT/HoTT filter wants coherence across routes. So the loss should include cross-chart transport consistency:

- transport contours and have them align with the next-frame contours,
- transport regions and have them align with next-frame regions,
- transport landmarks and have them align with next-frame landmarks.

When transport succeeds for some charts and fails for others, that disagreement becomes part of the model's exported coherence trace (not something it hides). That is how we avoid "pretty but wrong" continuity.

Finally, the transport objective must include *strain*. The model should output a measure of how hard the transport was—residual mismatch, inconsistency between charts, sensitivity to small perturbations. Strain is what supports boundary-event detection later.

So transport training is not only a self-supervised trick. It is the backbone that converts video perception from per-frame labeling into path-based identity.

6.3 Witness prediction: learning when two parses are equivalent up to re-factorization

The distinctively HoTT-flavored claim in this project is not "identity is transport" (many people already agree). It is that discrete refactorizations should be treated as ordinary chart changes—meaning the system should learn when two different decompositions are *equivalent* in the sense relevant to the task.

This requires a concept of a **witness** that mediates between parses.

Operationally, suppose the model produces two different parses of the same region:

- Parse A: a contour-dominant decomposition with certain junctions.
- Parse B: a region-dominant decomposition with a different boundary placement.
- Or in tracking: Parse A says instance split; Parse B says instance merge.

A witness is an exported object that says: "these parses are compatible under a refactorization mapping." In practice, that mapping can be:

- correspondence between cell elements (which junction matches which, which boundary corresponds to which),
- a refinement/coarsening relation (Parse A refines Parse B),

- a homotopy-like deformation (one boundary can be continuously deformed to another without crossing high-confidence barriers),
- a causal explanation (occlusion explains why one chart lacks elements present in another).

Learning witnesses means training the model to output:

- candidate correspondences between elements of parses,
- a score indicating whether a coherence-preserving refactorization exists,
- and an explanation trace of what constraints are preserved.

The training signal can come from controlled data where the ground truth refactorization is known (synthetic occlusions, simulated viewpoint changes, parametric shape families), and from weak supervision generated by Route A (algorithmic incidence graphs and their derived compositions can seed witness targets).

Crucially, witness prediction must include failure modes. Sometimes two parses are not equivalent; one is simply wrong, or the scene truly changed. The model must learn to output "no witness" when the constraints cannot be reconciled.

This is where the CT/HoTT inspiration becomes a real differentiator: instead of forcing a single best parse, the model learns to manage a space of parses and to export the structure that relates them—or to explicitly state that no relation exists.

6.4 Anti-overtransport mechanisms: refusal and abstention as trained behaviors

Equivalence-first seeing has a characteristic pathology: **overtransport**. If a system is rewarded for maintaining continuity, it can hallucinate witnesses and "explain away" contradictions. Humans do this too: we rationalize. A CT/HoTT filter must not become a coherence laundering machine.

So anti-overtransport is not optional; it is part of the design.

There are several mechanisms that can be trained explicitly:

- **Abstention head**: a learned output that can say "unknown / cannot transport" with calibrated probability.
- **Refusal loss**: penalize confident wrong witnesses more than abstentions, especially in adversarial or boundary regimes.

- **Conflict-sensitive training**: include examples where continuity is impossible (true identity swaps, hard cuts, object disappearance), so the model learns that "no witness exists" is sometimes correct.
- **Constraint diversity**: ensure coherence is checked across independent routes so the model cannot satisfy all constraints by a single degenerate trick.
- **Adversarial counterexamples**: generate cases where superficial cues encourage false continuity (lookalikes, deepfakes, repeated textures) and reward the model for refusing.

The ethical dimension appears already at the technical level: a model that always claims a witness is a proto-oracle. A safe model must be trained to treat refusal as success in the right conditions.

This also has a perceptual benefit: refusal reduces snaps. Snaps often occur when a system must suddenly reverse a wrong commitment. If it abstains earlier, the transition is smoother and more honest.

6.5 Practical training data: synthetic perturbations, controlled occlusion, lighting sweeps, edit provenance

Route B's training data should be designed to teach chart validity, transport, and boundary zones. That means the data must contain controlled transformations where we know what should remain invariant and what should not.

A practical curriculum includes:

- **Synthetic perturbations**: controlled geometric transforms (rotation, scale), photometric transforms (exposure, color temperature), blur/compression artifacts, and localized warps. These teach chart-specific fragility (texture charts fail under blur; contour charts fail under heavy noise).
- **Controlled occlusion**: masks that hide parts of objects/faces in structured ways (lower-half occlusion, random patches, moving occluders). These teach the model to maintain transport when possible and refuse when occlusion destroys constraints.
- **Lighting sweeps**: systematic changes in lighting direction and hardness. These teach that some cues are not identity-bearing and that chart choice must adapt.
- **Viewpoint sweeps and motion trajectories**: continuous rotations and translations to teach smooth transport and to create natural boundary zones where one chart becomes ill-conditioned and another takes over.

- **Edit provenance**: datasets that contain known edits (splices, deepfake-style manipulations, audio-video desync, frame interpolation artifacts). These teach the model to detect failure-of-lift: when local realism exists but constraint coupling fails over time.

Route A can be used here as a weak supervisor: it can generate candidate incidence structures, chart confidence estimates, and boundary-zone annotations that bootstrap training without requiring full human labeling.

The central idea is that training data must teach the model that "continuity is earned." Continuity is not a label; it is a witness that must survive perturbations and preserve constraints.

6.6 What the model exports: not just labels, but witnesses and coherence traces

The final deliverable of Route B must not regress to the usual outputs. If the model exports only a segmentation mask or a label, we have rebuilt object-first perception with better priors but the same epistemic flaw.

A learned CT/HoTT filter should export a structured percept artifact containing:

- **Chart renderings**: multiple decoders' outputs (contours, regions, landmarks, motion) with conditioning scores.
- **Transport objects**: predicted transforms and explicit transported correspondences between chart elements across time.
- **Witness objects**: mappings that relate competing parses (refinement/coarsening correspondences, deformation compatibility, "no witness" flags).
- **Coherence traces**: time-series and spatial maps of cross-chart agreement, with localized explanations of constraint failures.
- **Boundary-event annotations**: overlap zones where chart dominance shifts, with logged decisions when discretization is forced.
- **Abstention/refusal outputs**: explicit states where the model declines to assert identity or equivalence.

This exported artifact is what makes the model usable as more than a black box. It enables auditing, contestation, and training. It also enables human learning: XR-style display of chart changes and transport strain can teach users to perceive boundary zones as ordinary rather than as glitches.

Route B, done correctly, is therefore not "AI replaces the metaphysics." It is "AI learns better charts and better witnesses under a discipline that keeps them visible." The learning gives power; the exported structure gives accountability. Together they aim at the paper's core goal: a converter that makes refactorization ordinary by representing continuity as something witnessed—transported, checked, and shown—rather than merely asserted.

7. Representation Choices and Their Consequences

A CT/HoTT percept filter is not one algorithm. It is a commitment to a certain kind of exported structure: charts, witnesses, coherence traces, boundary zones. But those outputs can be realized with different internal representations, and the choice matters. Representation is not neutral. It determines what "continuity" is cheap to express, what "refactorization" looks like, and where snaps tend to occur.

This section compares four representational families—mesh-first, implicit-field, path-first, and hybrids—and then specifies a principled chart-selection rule. The goal is to make tradeoffs explicit so that "CT/HoTT-style" does not become a vague aesthetic label.

7.1 Mesh-first outputs: explicit parts, explicit remeshing, explicit snap risk

A mesh-first representation discretizes early and proudly. It turns the world into vertices, edges, faces, and adjacency. In images, the closest analog is a hard segmentation with explicit boundaries and a graph of regions; in 3D, it is literally a polygon mesh; in video, it is a tracked set of explicit parts.

The virtues of mesh-first are obvious:

- **Interpretability**: the representation is concrete and inspectable.
- **Computation**: discrete structures enable efficient algorithms for many tasks.
- **Communication**: meshes and graphs serialize well; they are shared references.
- **Downstream use**: robotics, physics simulation, and CAD-like tasks often want explicit geometry.

But mesh-first has a signature cost: **refactorization becomes an event.** If your world is vertices and faces, then changing vertex valence or splitting/merging faces is not just a parameter update—it is a topological edit. Those edits feel like snaps because the representation itself changes its combinatorial type.

This is exactly the cube–octahedron stress test. In a mesh-first story, “3 faces at a vertex vs 4” is primitive; transitioning between them requires a remeshing event, and remeshing is naturally discontinuous in the combinatorial description even if the underlying shape varies continuously.

So mesh-first is compatible with CT/HoTT filtering only if the system explicitly treats meshes as *charts*—regime-dependent discretizations—and exports boundary zones where remeshing is expected. Without that discipline, mesh-first encourages ontology drift: the mesh becomes the object, and remeshing becomes “the object changed.”

In short: mesh-first is excellent for late-stage summaries and for action interfaces, but high snap risk makes it a poor sole substrate for relation-first perception.

7.2 Implicit fields: continuous substrates with late discretization

Implicit-field representations take the opposite stance: keep the world continuous as long as possible. In vision, “fields” can mean many things: signed distance functions, occupancy probabilities, neural implicit surfaces, continuous orientation/curvature fields, or dense latent feature fields.

The virtues here align naturally with our aim:

- **Continuity is native**: small changes in input produce small changes in the field.
- **Refactorization is deferred**: you can derive discrete objects later by thresholding or extracting level sets.
- **Boundary zones are representable**: uncertainty and chart overlap can live as gradients and mixed regimes rather than as discrete edits.

Implicit fields tend to reduce snaps because they do not need to “change the data structure” when the best discrete summary changes. A boundary tube can thicken and shift without a topological edit. A region can gradually lose confidence without abruptly splitting. In video, a transport field can weaken gradually, producing strain rather than an ID flip.

The cost is also clear:

- **Interpretability can be indirect**: a field is not a face/edge/vertex; it is a substrate from which those are extracted.
- **Downstream demands**: many applications still require discrete commitments (masks, tracks, landmarks).
- **Calibration burden**: continuous fields can look confident everywhere unless they are explicitly trained/engineered to expose uncertainty.

From the CT/HoTT perspective, implicit fields are a strong candidate for the *core substrate*. They allow us to treat charts as different ways of slicing or parameterizing the same underlying field, and they make "chart change" an ordinary phenomenon: the field remains; only the extraction changes.

So the guiding principle becomes: fields for perception, meshes for reporting—unless your task requires explicit combinatorics continuously.

7.3 Path-first state: identities as transport chains over time

A third representational family is not primarily spatial at all. It is temporal and relational: the core state is a set of *paths*—transport chains—linking percepts across time and across transformations.

In standard video systems, "tracking" already gestures at this, but usually in a thin form: track IDs and bounding boxes. A path-first state is thicker:

- each entity is represented as a transport history, not as a static template;
- the state stores not just "where it is," but "how it has been transported";
- strain and failures are part of the state.

The virtue of path-first representation is that it makes our key philosophical move operational: identity is not equality; identity is a witnessed path. The output of the converter becomes a set of explicit witness objects—composable transports—so the system can explain continuity claims.

Path-first representations are especially powerful for:

- occlusion handling (identity persists through uncertainty windows),
- deepfake/edited video detection (transport failures show up as witness breaks),
- boundary zones (strain increases before discrete decisions flip).

The costs and risks:

- **State explosion**: storing rich path histories can be expensive.
- **Loop closure errors**: long transport chains can drift unless periodically re-anchored.
- **Overtransport temptation**: a system that "wants" continuity can hallucinate path extensions unless refusal is trained/engineered.

In practice, path-first state pairs naturally with implicit fields. The field provides local structure; the path state provides global continuity. Together, they make “lift succeeds” a measurable condition: can we transport the local structure forward and have it agree with new evidence?

7.4 Hybrid stacks: continuous core, discrete summaries on demand

The most plausible architecture for a CT/HoTT percept filter is hybrid:

- **Continuous core**: implicit fields (appearance, motion, geometry) that evolve smoothly.
- **Relational continuity layer**: path-first state storing transport witnesses and strain.
- **Discrete chart summaries**: meshes/masks/landmarks extracted when needed, treated as charts rather than as reality.
- **Commit layer**: a downstream module that produces single decisions for action, with logged justification and abstention options.

This hybrid stack matches the paper’s metaphysics: perception should not be forced to discretize early, but action often requires discrete commitments. The trick is to keep those commitments downstream and reversible, and to export the richer state so that revisions are seen as chart changes.

In this hybrid view, a “snap” becomes an error of interface design, not an unavoidable fate:

- If the user only sees the commit layer, they will perceive snaps.
- If the user sees the chart family plus boundary zone annotations, they will perceive transitions.

Hybrids also allow graceful degradation. When compute is limited, the system can reduce the number of charts it maintains, or shorten the path-history horizon, while still preserving the audit trail structure.

So the hybrid claim is: don’t pick one representation and worship it. Use a continuous substrate for stability, a path layer for identity, and discrete summaries as optional, logged slices.

7.5 How charts are chosen: stability, conditioning, and action constraints

If multiple charts coexist, the system must decide which chart to privilege at any moment, and how to transition between charts without snaps. The rule cannot be “whichever is most confident,” because confidence can be miscalibrated. The rule must be grounded in conditioning and coherence.

A principled chart-choice mechanism uses three inputs:

- **Stability / conditioning**: how well the chart's local model fits current data, with residuals and sensitivity measures. A chart is ill-conditioned if small perturbations produce large changes in its output.
- **Cross-chart coherence**: how well the chart agrees with other charts and with transport history. A chart that fits locally but disagrees globally should be discounted.
- **Action constraints**: what discrete commitments are required by the downstream task (e.g., robot grasp requires a hard boundary; surveillance requires an ID decision; editing requires consistent keypoints).

Given these, chart selection should behave like a control system, not a winner-take-all vote:

- maintain a weighted mixture of charts in ordinary regimes;
- detect rising strain as conditioning declines and coherence dips;
- enter an overlap zone where weights shift gradually;
- only commit to a new dominant chart when it becomes stable and coherent;
- if action requires a hard decision during an overlap zone, commit with an explicit log entry and an explicit "fragile" marker.

This is the operational meaning of "boundary events as zones." A boundary event is not one frame. It is the interval during which the system's preferred chart changes because the old chart becomes ill-conditioned and the new chart becomes stable. In that interval, the system should surface uncertainty rather than hide it.

Finally, chart choice must respect refusal. If no chart is well-conditioned and coherent, the system should abstain from strong claims. This is not a weakness; it is the hallmark of a non-oracular percept.

So representation choices and chart-choice rules are inseparable. Mesh-first encourages snaps unless softened by a field substrate. Implicit fields reduce snaps but require careful extraction discipline. Path-first identities make continuity explicit but must avoid hallucinated transport. Hybrids can reconcile these, provided chart choice is driven by conditioning and coherence, and provided action commits are logged rather than mistaken for reality.

8. The CT/HoTT Filter as a Visual and Video Style

Up to this point we have described a new output contract: charts, witnesses, coherence traces, boundary events. But unless the output can be *seen*—in the ordinary sense of a human being inspecting an image or video—then the converter will collapse into a private internal representation again. The practical wager of this paper is that CT/HoTT ideas are not only implementable; they can become a *visual style* that makes representational transitions legible.

This section treats the CT/HoTT filter as a rendering problem: how to display multiple decompositions without clutter, how to show transport without turning the screen into spaghetti, how to mark boundary zones without implying catastrophe, how to raise alarms without producing "oracle vibes," and how to give the user controls that support audit rather than passive consumption.

The style goal is: **make refactorization ordinary, and make reasons visible.**

8.1 Chart overlays: showing competing decompositions without confusion

The first UI problem is psychological: people are trained by ordinary vision systems to expect one answer. If we show multiple charts, the viewer may interpret that as indecision or incompetence. The style must therefore make multiplicity feel principled.

A workable approach is to treat charts like map layers:

- **One chart is dominant by default**, chosen by conditioning and coherence.
- **Other charts are available as overlays**, each with an explicit role label: "Contour chart," "Region chart," "Motion chart," "Landmark chart."
- **Overlays are soft, not binary**, and they use transparency to communicate partial validity: thicker or more opaque means "more stable here," thinner or faint means "less stable."

Key to avoiding confusion is enforcing a consistent hierarchy of visual grammar:

- Contours should render as **tubes** (not lines) so uncertainty is visible.
- Regions should render as **soft partitions** with boundary fuzz, not hard masks.
- Junctions should render as **fields** or small neighborhoods, not as single pixels.
- Each chart should carry an on-screen **conditioning indicator** that can be localized: "Contour chart stable: high," but also "Contour chart unstable near this occlusion edge."

A second anti-confusion device is **gating by locus**. Instead of showing full-frame overlays all the time, the UI can show overlays only in the region the user is inspecting or only where charts disagree. This makes multiplicity a targeted diagnostic, not a permanent distraction.

Finally, chart overlays must never pretend they are "different opinions about the world." They are different coordinate choices. The UI should reinforce this by allowing the user to *flip the chart basis* while keeping the percept anchored: the same locus remains selected, the same time index remains selected, and only the descriptive layer changes.

The desired qualia is: "I am changing my description, not changing the object."

8.2 Transport visualization: flows, arrows, and stability/strain traces

Transport is the signature of this style. Without transport rendering, "identity as witness" remains hidden, and the system becomes another tracker with a new vocabulary.

Transport visualization must solve a classic problem: if you draw all correspondences, you get an unreadable tangle. The CT/HoTT approach insists on compositional structure, so the visualization should emphasize *composable paths* rather than raw point matches.

Three levels work well:

- **Macro transport**: show object-level or region-level transport as a small number of arrows or ribbons, one per entity, connecting its position across frames.
- **Meso transport**: within a selected entity, show a sparse set of representative correspondences (e.g., key anchors) rather than dense flow everywhere.
- **Micro transport**: when the user pins a locus, show local transport details at that locus only.

The second requirement is to visualize **strain**, not just motion. Motion is not the point; coherence is. Strain is what distinguishes "smooth transport" from "forced continuity."

Strain traces can be rendered as:

- thickness changes in transport ribbons,
- a small stability meter near the selected entity,
- localized heat around points where transported structure fails to align with new evidence,
- time-series traces showing strain rising before boundary events.

This is where the “lift succeeds” idea becomes perceptible. The UI can allow the user to toggle between:

- “render at time t, then transport forward”
- “render at time t+1 directly”

and show the difference as an error overlay. When the lift is good, the difference is small and diffuse. When it fails, the difference becomes localized and structured. That difference overlay is the visual analog of a failed commutative square, but presented without requiring the user to know that term.

Transport visualization should also respect refusal: if the model cannot transport confidently, it should render a broken ribbon or a faded region with an explicit “witness unavailable” marker. This avoids the most dangerous vibe: a clean arrow drawn confidently through uncertainty.

8.3 Boundary visualization: critical zones as ordinary, not alarming

Boundary events are where ordinary systems snap. Our style goal is to make boundary zones feel as normal as turning your head and briefly losing fine detail.

The first rule is: **render boundaries as zones, not moments.** In video, this is a time window; in images, it is a spatial neighborhood. The UI should show:

- an “overlap band” where chart A and chart B both have partial validity,
- a gradual handoff of dominance,
- and explicit reasons: “Contour chart ill-conditioned due to blur,” “Region chart ill-conditioned due to shadows,” “Motion chart uncertain due to occlusion.”

The second rule is: **avoid alarm language for ordinary boundaries.** A boundary zone is not necessarily an error; it is a representational transition. The UI should visually distinguish:

- ordinary boundary zones (expected transitions due to viewpoint/occlusion),
- suspicious boundary zones (inconsistencies that do not match expected physics),
- and hard discontinuities (cuts, insertions, or missing frames).

This matters ethically and psychologically. If the UI treats every boundary as a warning, users will ignore warnings. If the UI treats boundaries as ordinary but marks when they are *abnormal*, then warnings retain meaning.

A concrete rendering pattern is a **timeline band** beneath the video with colored or textured segments:

- “stable regime” segments,
- “boundary zone” segments,
- “abstain / witness missing” segments,
- “coherence break” segments.

The user can scrub into those segments to inspect them. The UI’s story is: this is the normal topology of perception over time—stable, transition, stable—not a sequence of abrupt truths.

8.4 Coherence alarms: what is flagged and how it is explained

Coherence alarms are the filter’s integrity mechanism. They must be designed to avoid two failures:

- over-alarming (noise that makes the tool unusable),
- and under-explaining (black-box warnings that invite oracle worship).

A coherence alarm should be raised only when a *structural disagreement* occurs—meaning independent routes disagree in a way that cannot be reconciled by known boundary conditions.

Examples of alarm-worthy signals:

- transport predicts a motion-consistent deformation, but the directly rendered next frame violates that deformation in a structured region,
- region boundaries and contour boundaries diverge in a way that cannot be explained by blur/lighting/occlusion confidence,
- a witness is claimed between two parses, but the induced composition fails (the “derived incidence” does not match direct evidence),
- a deepfake-like sequence looks plausible per-frame but fails in cross-frame constraint coupling.

For each alarm, the tool should provide an explanation in the only form that matters here: **show the conflicting constraints.** This can be done with a small “constraint card” that contains:

- which charts disagreed (e.g., motion vs landmarks),
- where the disagreement lives (spatial highlight),
- what the system expected under the claimed witness,

- what was observed instead,
- and what the system did: shift charts, abstain, or commit with a fragility tag.

This explanation style keeps the tool honest. It does not say “this is fake.” It says “these constraints did not cohere; here is where; here is why.” That supports contestability and prevents the tool from becoming a verdict machine.

Importantly, coherence alarms should have degrees. Many disagreements are mild and expected at boundary zones. The UI should show severity as a function of both magnitude and context: the same magnitude of disagreement is less alarming inside a known boundary zone and more alarming in a stable regime.

8.5 User-facing controls: scrub time, pin a locus, compare parses

The CT/HoTT filter’s value is realized through interaction. A static rendering cannot convey chart changes and transport strain. The core controls should support auditing workflows:

- **Scrub time with regime awareness**: as the user scrubs, the UI indicates which regime they are in (stable vs boundary vs abstain), and allows jumping between boundary events.
- **Pin a locus**: the user clicks a point or region, and the UI locks onto that locus through time, showing transport and chart validity there. This turns “watching video” into “tracking a structure.”
- **Compare parses**: the user can toggle between chart renderings at the same time point, or between “transported rendering” and “direct rendering,” with difference overlays.
- **Inspect witnesses**: a panel shows the active witness object for a selected entity, including its strength, strain, and failure points.
- **Replay decisions**: if a discrete commitment was forced (mask selection, ID choice), the user can open the decision log and see the alternatives that existed at that moment.

A crucial control is **abstention visibility**. The UI must make “I do not know” actionable: when the system abstains, the user can inspect what constraint prevented witness construction and can decide whether to accept the abstention, provide additional data, or lower the claim level.

In effect protect, the UI should communicate (without preaching) a hierarchy:

- perception exports structures and uncertainties,
- decisions are downstream and logged,
- truth is not guaranteed by coherence.

That last point is the moral center of the style. The CT/HoTT filter can be visually seductive—flows, charts, and "commutativity-like" stability can feel like deep correctness. The UI must therefore preserve humility: render confidence and strain together, make boundary zones ordinary, and make alarms explain their reasons rather than issuing verdicts.

If we do that, this becomes a new visual genre: not "psychedelic geometry," but "structural perception." The viewer learns to see transitions as chart changes, to see continuity as witnessed transport, and to treat the world as stable even when descriptions refactor—because the refactorization is what the tool was built to show.

9. Benchmarks and Experiments

A CT/HoTT percept filter is not justified by elegance alone. Its claims are behavioral: fewer snaps, witnessed continuity, boundary zones made explicit, and calibrated refusal instead of hallucinated equivalence. Those claims require benchmarks that measure not only "did the model get the label right," but "how did the model behave while deciding." In ordinary leaderboards, many of the phenomena we care about are invisible: a tracker that flips IDs twice but ends up correct at the end may score fine, while a system that openly abstains and flags boundary zones may score worse even though it is epistemically safer.

This section proposes an experimental program designed to evaluate the output contract itself. The unit of measurement is not only accuracy, but the geometry of revisions: whether the system reweights continuously, whether transport strain rises before changes, whether boundary zones are visible, and whether refusals are calibrated.

9.1 Invariance profiles: occlusion, lighting, and viewpoint sweeps

The first benchmark family measures *invariance profiles*. The question is not: can the system recognize a face under a single harsh condition? The question is: how does the system's percept evolve as conditions change smoothly? Does it drift smoothly, does it enter an overlap zone, does it abstain when necessary, and does it recover without snaps?

A practical design is to construct controlled sweeps along three axes:

- **Occlusion sweeps**: progressively cover an object or face with an occluder whose area increases smoothly (0% to 80%), and whose motion is controlled (static, sliding, rotating).
- **Lighting sweeps**: vary exposure and light direction continuously (e.g., rotate a virtual light source around a face model, or use controlled real lighting rigs).

- **Viewpoint sweeps**: rotate the object/face in a controlled manner, including passing through angles where standard landmarks become ill-conditioned (profile views, extreme pitch).

For each sweep, we record the system's exported artifact over time:

- chart conditioning curves (which charts are stable when),
- transport witness strength and strain,
- coherence scores across routes,
- boundary-event annotations,
- abstentions.

The output is not one number but an "invariance signature" plot: as the sweep parameter changes, does the system show smooth reweighting and honest boundary zones, or does it hold a brittle commitment until it snaps?

The important experimental control is *smoothness of the stimulus*. If the input changes smoothly but the output changes discontinuously without boundary-zone explanation, that is evidence of snap-prone representation. If the output transitions through overlap zones with rising strain and explicit chart handoff, that supports the CT/HoTT claim.

9.2 Ambiguity handling: non-collapse and continuous reweighting vs flips

The second benchmark family targets a distinct behavior: how the system handles ambiguity. Many failures in perception are not about incorrectness but about premature collapse. The system could have admitted uncertainty, retained multiple hypotheses, and waited for disambiguating evidence, but instead it chose a single parse and then flipped later.

We therefore design tasks where ambiguity is intrinsic and then gradually resolved:

- **Two similar identities crossing** (video): two faces with similar appearance cross paths with partial occlusion; later frames provide disambiguating cues.
- **Split/merge topology** (segmentation): an object moves behind a thin obstruction causing an apparent split, then re-emerges.
- **Viewpoint-induced symmetry**: objects that look identical under certain angles but differ under others.

The evaluation is not simply "was the final identity correct?" It is:

- Did the system maintain multiple candidate transports during ambiguity?

- Did it reweight continuously as evidence accumulated?
- Did it flag a boundary zone during the ambiguous interval?
- Did it abstain instead of committing and flipping?

This is where our output contract becomes measurable. A CT/HoTT filter should show that ambiguity is a state, not a failure. It should treat the ambiguous interval as an overlap zone with soft incidence and multiple chart candidates, and it should only collapse to a single explanation when conditioning and coherence justify it. A standard object-first model tends to collapse early and then flip.

So the metric here is a "collapse curve": how quickly the model collapses to a single parse as ambiguity begins, and how violently it flips when evidence changes. Success is delayed collapse, continuous reweighting, and a logged decision when commitment is required.

9.3 Synthetic media sensitivity: constraint-consistency vs artifact spotting

Deepfakes and synthetic edits expose a particular weakness of ordinary perception systems: many are trained to spot artifacts, but a skilled generator can reduce artifacts while still violating deeper constraints. A CT/HoTT filter aims to detect synthetic media primarily by *constraint inconsistency*, not by superficial artifacts.

So we propose benchmarks that separate these regimes:

- **Artifact-light synthetic video**: high-quality generation where per-frame realism is strong.
- **Constraint-violating edits**: subtle temporal inconsistencies (microexpressions that do not match head motion, eye-blink dynamics inconsistent with gaze changes, lip motion misaligned with face muscle deformation).
- **Splice and cut tests**: insert short segments from different sources with careful color matching to suppress obvious seams.

The benchmark asks: does the system's coherence trace detect "failure of lift"?

A CT/HoTT-style system should exhibit:

- increased transport strain even when frames look plausible,
- localized cross-chart disagreements (e.g., motion chart says one deformation, landmark chart says another),
- boundary events that do not align with expected physical causes (no occlusion, no lighting shift, but a chart handoff occurs anyway),

- explicit refusal to assert witnesses when constraints cannot be reconciled.

By contrast, artifact spotters may be fooled when artifacts are minimized. This benchmark therefore rewards a different competence: not “I found a weird pixel pattern,” but “these constraints do not compose.”

The evaluation must remain epistemically modest: coherence breaks do not prove fakery; they indicate inconsistency. The benchmark’s goal is to test whether inconsistency is detected and explained, not whether a verdict is issued.

9.4 The polyhedron-movie task: graded incidence perception and chart overlap

The polyhedron-movie task is the conceptual anchor of this project: a controlled visual sequence where discrete incidence summaries change, yet the underlying shape variation is smooth. This task is designed to measure whether the system can represent “3→4” type switches as ordinary chart transitions rather than as ontological snaps.

We construct a dataset of parameterized polyhedral morphs and truncations (synthetic rendering is acceptable and desirable because ground truth is controllable). Each sequence is a smooth path through a shape family that includes known “critical zones” where discrete combinatorial summaries change:

- face counts and vertex valences change through truncation/rectification,
- edges may persist while incidence relations refactor,
- duality-like transitions can be staged by varying which chart is dominant (face-centric vs vertex-centric).

The converter is evaluated on its ability to output:

- graded incidence (soft attachments) near critical zones,
- chart overlap where alternative decompositions are both plausible,
- boundary-event zones that align with the known critical intervals,
- and coherent transport of structure across the sequence.

We can define a specific “graded incidence” test: near the critical zone, does the system ramp the participation of the “emerging face” smoothly before it becomes fully discretized, or does it remain blind and then snap?

In addition, we can test duality sensitivity: does the system’s incidence representation make it easy to render both primal and dual views, and to show the “binocular” co-presence discussed

earlier? Even if the system does not "know duality," it can still output incidence structures that admit dual readings.

This task is not meant to be a novelty stunt. It is a calibrator: it forces the system to confront the exact phenomenon that motivated the paper—discrete incidence change under smooth deformation—and it yields clean, interpretable measures.

9.5 Metrics: time-to-stability, reversal geometry, coherence AUC, calibrated abstention

Finally, we need metrics that match our claims. Standard top-1 accuracy does not measure snaps. We propose a small suite of metrics, each tied to a behavioral signature.

- **Time-to-stability**: after a perturbation or entry into a boundary zone, how long does it take for the system's chart weights and identity transports to stabilize? A snap-prone system may stabilize instantly—but by collapsing prematurely—and then later flip. A coherence-first system may take longer but should do so with continuous reweighting and explicit uncertainty. Time-to-stability should be reported alongside an "instability penalty" that counts post-stabilization flips.
- **Reversal geometry**: measure how the system behaves when a sweep parameter is reversed. For example, occlude and then un-occlude, rotate left and then rotate back. A good coherence-first system should show hysteresis only when justified by conditioning; a snap-prone system often shows inconsistent hysteresis: it follows one path forward and a different path backward with abrupt flips. "Reversal geometry" quantifies the discrepancy between forward and backward trajectories in chart-space and witness-space.
- **Coherence AUC**: treat coherence as a detection signal for correctness. When ground truth identity/persistence exists, does higher coherence correlate with correctness? Coherence AUC measures how well coherence separates "likely correct continuity" from "likely incorrect continuity." This tests whether coherence is meaningful or merely decorative.
- **Calibrated abstention**: measure refusal behavior under ambiguity. A key metric is selective risk: as the system abstains more, does its error rate on non-abstained cases drop appropriately? More specifically, do abstentions concentrate in boundary zones and truly ambiguous cases, or do they occur randomly? Calibrated abstention can be evaluated with reliability curves and with a "boundary alignment score" (fraction of abstentions that occur inside known or detected boundary zones).
- **Snap rate and snap explainability**: define a snap operationally as an abrupt change in a discrete summary (ID, segmentation topology, incidence census) without a preceding

overlap zone of sufficient length or without rising strain/coherence warning. We can then measure snap frequency and, importantly, how many discrete changes were accompanied by a logged boundary event. The goal is not zero changes; the goal is zero unaccounted changes.

Together these metrics shift evaluation from "did you guess right" to "did you behave honestly under change." A CT/HoTT percept filter is trying to upgrade the epistemic behavior of vision systems: to treat chart changes as ordinary, to export witnesses when continuity is claimed, and to show boundary zones rather than hiding them. These benchmarks and metrics are designed to make those claims falsifiable in the way that matters: in the geometry of transitions, not only in the final answer.

10. Applications

The CT/HoTT percept filter is not a new "model of reality." It is a new *artifact of perception*: chart families, transport witnesses, coherence traces, and boundary zones rendered in a way humans can inspect. The applications therefore cluster around situations where ordinary outputs (labels, masks, tracks) are not enough—either because the cost of snap-based error is high, or because the user needs reasons rather than verdicts, or because the goal is education and experience rather than automation.

This section outlines five application families. Each is framed in terms of what the CT/HoTT filter uniquely exports: visibility of refactorization, witnessed continuity, and explicit uncertainty.

10.1 XR training: can humans learn chart-change qualia?

A surprising application is pedagogical and phenomenological: use XR to train humans to experience representational refactorization as ordinary.

Humans already have weak analogs of chart change. We tolerate appearance changes under lighting. We maintain identity through motion. But we also exhibit characteristic snaps: ambiguous figures flip; occluded objects "pop" in and out; we make hard commitments and then revise them abruptly. The CT/HoTT filter provides a new form of feedback: it externalizes overlap zones and transport strain that our perceptual system normally hides from consciousness.

In XR, this can become a training loop:

- The user views a scene (real or synthetic) while the CT/HoTT filter overlays chart validity regions and boundary zones.

- The user is asked to predict when an ID or segmentation would “flip” under ordinary systems.
- The system then shows the overlap zone and the coherence trace that explains why a flip is fragile.
- Over repeated exposure, the user may learn a new internal habit: hold multiple parses longer; expect chart tradeoffs; experience boundary zones as natural.

The measurable question is exactly the one you raised earlier: can humans acquire “chart-change qualia”? Not mystical insight, but a stable cognitive skill: interpreting representational transitions as coordinate changes rather than as reality breaks.

If successful, this would have downstream benefits. Operators in surveillance, medical imaging, and editing could become less vulnerable to snap-induced overconfidence. They would learn to interpret “seems obvious” as “a chart is dominating right now,” and to check whether coherence supports it.

So XR is not merely a visualization medium; it is a training environment for perception discipline.

10.2 Scientific imaging: audit trails for segmentation and identity claims

Scientific imaging is full of hidden snaps. In microscopy, materials imaging, and medical scanning, segmentation and tracking are often used as if they were facts. But practitioners know the truth: segmentation is a model, not a revelation. Small changes in preprocessing or thresholds can split a cell, merge two nuclei, or change the measured boundary of a lesion. These are refactorizations, and they matter.

The CT/HoTT filter’s main value here is the audit trail:

- multiple chart renderings (texture/edge/shape/motion) rather than one segmentation;
- conditioning measures indicating where each chart is reliable;
- explicit boundary zones where segmentation is unstable;
- identity as transport for time-lapse microscopy (cell tracking) with witness strength and strain;
- logged discretization choices and their alternatives.

This changes how results can be reported. Instead of publishing a single mask, one can publish a mask plus a boundary-zone overlay that indicates where the boundary is uncertain and how that uncertainty was handled. Instead of claiming “cell A divided at time t,” one can show a

transport chain whose strain rises and then splits, with evidence that the split is structural rather than an artifact.

In short, scientific imaging is where “coherence traces” are not luxury; they are the difference between measurement and impression.

10.3 Accessibility: stabilizing recognition under adverse conditions

Accessibility applications benefit from a different feature: graceful behavior under adverse conditions. Many assistive vision systems struggle because they must provide crisp, user-facing answers (“that is your friend,” “that is a door”) in conditions where perception is inherently ambiguous: low light, motion blur, partial occlusion, camera shake, or unusual viewpoints.

A CT/HoTT filter can improve accessibility not by hallucinating certainty, but by stabilizing user experience:

- maintain continuity with transport when conditions degrade rather than flipping IDs;
- show boundary zones as “uncertain regions” and guide the user to reframe;
- provide graded, contestable suggestions rather than brittle assertions;
- abstain in a way that is informative (“I cannot verify identity right now; motion constraints conflict”) rather than silent failure.

For a user depending on assistive tech, the worst failure mode is snap: “this is your friend” one moment, “unknown person” the next, without explanation. A coherence-first system can maintain a stable interaction: it can preserve the hypothesis while marking its fragility, and it can request additional data (a different angle, more light) in a targeted way.

So accessibility is not only about higher accuracy. It is about an epistemically stable interface: a system that behaves like a careful assistant rather than like a jumpy classifier.

10.4 Forensics: contestable signals rather than verdict machines

Forensics is where the “oracle temptation” becomes acute. People want machines to deliver verdicts: “fake,” “real,” “this person,” “that person.” But that is exactly where systems become socially dangerous, because a confident verdict has moral and legal weight.

The CT/HoTT filter supports a different posture: provide contestable signals.

Instead of a verdict, the system exports:

- where constraints cohere and where they break,
- what transport witness exists (or does not),

- what boundary events occurred and why they are suspicious or ordinary,
- and how alternative parses remain viable.

This supports an adversarial process. A forensic analyst can inspect the witness, challenge the assumptions, and present the coherence trace as evidence rather than as authority. The system becomes a microscope for inconsistencies, not a judge.

This is particularly important in synthetic media analysis. Many deepfake detectors are brittle and can be gamed; they also tend to output a binary verdict that invites misplaced trust. A constraint-consistency tool can be more robust and more honest: it can show that certain temporal couplings fail, while still acknowledging that incoherence is not proof of intent or origin.

So the application is not "AI says fake." The application is "here is the structural reason this sequence fails to compose under expected constraints." That is a form of evidence compatible with humility.

10.5 Creative tooling: "category picture" aesthetics as an output genre

Finally, there is an application that returns to the origin of the idea: art. Your earlier "incidence category picture" for cube–octahedron duality treated structure as something that can be rendered onto a page without lying. The CT/HoTT filter extends that into a new visual genre for images and video: structural perception as aesthetic.

Creative tooling can use the filter as an effect—not an LSD-like hallucination, but an honest overlay of relational structure:

- chart overlays as layered textures,
- transport ribbons as calligraphic motion trails,
- coherence heatmaps as "truth gradients,"
- boundary zones as glowing transition bands,
- witness objects as visible correspondences.

This yields a style that is both mathematically motivated and visually legible: the image is not replaced; it is annotated by its internal structural negotiations. A portrait could show not only a face but its chart family: how identity is maintained under expression changes, how landmarks drift, how occlusions threaten transport, how the system re-describes without rupture.

In other words, the "category picture" moves from static polyhedra to dynamic perception. This could become a serious genre: an art form that renders the *audit trail of seeing*.

There is also a practical creative benefit: editors and VFX artists often fight temporal incoherence. A CT/HoTT filter that visualizes transport strain and boundary zones could be used as a debugging layer during editing: not merely to stylize output, but to locate where continuity will break.

So the creative application is not an afterthought. It is a proof of concept that the percept artifact is genuinely a new kind of object: a visualizable structure that sits between raw imagery and final labels, and that can be used for understanding, auditing, training, and aesthetics.

Across all five applications, the unifying theme is the same: a CT/HoTT percept filter is valuable where *reasons matter*, where *transitions matter*, and where *uncertainty must be visible*. It is a proposal to make perception less like a single-answer oracle and more like a disciplined witness—one that shows its work.

11. Cognitive Tradeoffs, Limits, and Failure Modes

A CT/HoTT percept filter is, by design, more honest than an object-first labeler: it carries multiple charts, keeps ambiguity alive, exports witnesses, and surfaces boundary zones. But honesty has costs. It is more computation, more data to display, more opportunities to confuse the user, and more ways to fail in subtle "too clever" directions. This section names those costs and failure modes explicitly, because a system that claims to reduce snaps can easily become something worse: a coherence machine that manufactures persuasive narratives.

The stance here is pragmatic: the goal is not maximal structure, but the right structure exported at the right moments, with refusal as a first-class behavior.

11.1 Cost profile: compute, latency, and attentional load

The first cost is computational. Multi-chart perception is not free. Even Route A requires parallel field computations, incidence construction, coherence scoring, and boundary-zone tracking. Route B adds learned decoders and witness prediction. Three specific cost dimensions matter:

- **Compute cost**: Maintaining multiple charts multiplies inference and post-processing. Transport objectives (especially dense ones) add a heavy term. Coherence scoring across routes adds cross-comparisons that scale with chart complexity.
- **Latency cost**: If the filter is used in real-time applications (XR, assistive tech), it must produce a stable percept with low delay. But boundary-zone detection often benefits from temporal context (looking a few frames ahead) to distinguish transient noise from a real regime change. That creates an inherent tension between immediacy and epistemic caution.

- **Memory/state cost**: Path-first identity requires storing transport chains, strain history, and alternate hypotheses. Long horizons improve stability but increase memory and risk drift.

There is also a human cost: **attentional load**. A rich percept artifact can overwhelm. If users must constantly parse overlays and meters, the tool becomes unusable outside research contexts.

So an honest CT/HoTT filter must include a *presentation policy*: it should default to minimal overlays, and it should surface structure primarily when something is unstable, contested, or consequential. In other words: most of the time, show the dominant chart and a simple "stability bar." When boundary zones appear, expand the view to show overlap and alternatives. The tool should behave like a debugger: quiet when things are stable, verbose when they are not.

This also suggests a tiered deployment approach:

- **Core mode**: one dominant chart + stability indicator + abstention when needed.
- **Inspect mode**: reveal competing charts, witnesses, and coherence trace.
- **Audit mode**: full event logs and constraint cards.

Without that tiering, the cognitive load will erase the benefit.

11.2 Pathologies: overidentification, undersegmentation, missed local salience

A relation-first system has characteristic pathologies that differ from object-first errors.

- **Overidentification**: Because the system is rewarded for transport and equivalence, it can become too eager to declare "same." It may stitch distinct entities into one identity chain, especially when they share global coherence cues (similar motion, similar texture). This is the continuity analog of overfitting: a model that prefers one big story.
- **Undersegmentation**: A coherence-first system may resist splitting because splits are refactorizations. It might smooth over genuine multiplicity, merging separate objects into a single region to preserve global coherence. This can be particularly damaging in scientific imaging (two cells merged), robotics (two objects treated as one), or social settings (two people merged into one track).
- **Missed local salience**: A system focused on global agreement can fail to notice small but important local events—an expression microchange, a small tool in a hand, a subtle lesion boundary. These are "low-mass" signals that may not strongly affect global coherence but matter for the task.

These pathologies are not hypothetical; they are the natural result of reweighting the perceptual primitive from objects to constraints. The fix is not to revert to object-first, but to include explicit counterweights:

- local novelty detectors that can interrupt coherence smoothing,
- split-permission mechanisms that allow segmentation to refactor when evidence strongly supports it,
- and task-conditioned salience channels: the system must know what kinds of local changes matter in a given domain.

In CT/HoTT terms, one might say: coherence is not the only invariant; certain local features are invariants-by-role for specific tasks. The filter must allow those roles to drive discretization when appropriate.

11.3 Coherence laundering: correlated routes that agree for the wrong reasons

The most subtle failure mode is **coherence laundering**: multiple routes agree, but they are not truly independent. They share a hidden bias, training artifact, or confound. The system then presents agreement as evidence, and the user is misled by a persuasive but circular justification.

This is a real danger because our system elevates agreement to a first-class signal. If agreement is fake, the whole epistemic benefit collapses.

Coherence laundering can happen in several ways:

- **Shared encoder bias** (Route B): if all chart decoders share the same latent state, they may agree because the shared representation is biased, not because the world supports the claim.
- **Data leakage**: training augmentations can unintentionally teach the model a shortcut that persists across charts (e.g., camera-specific compression signatures become "identity").
- **Cascaded dependence** (Route A): if later charts are derived from earlier ones (e.g., region segmentation uses edges that were already thresholded), then "agreement" is not cross-validation; it is self-confirmation.
- **Over-regularization**: coherence losses may force alignment even when disagreement would be appropriate, causing the model to suppress evidence of boundary zones.

The remedy is structural: enforce route diversity.

- In Route B, use partially decoupled pathways or separate encoders for some charts, so that agreement is meaningful rather than guaranteed.
- Use heterogeneous cues (appearance vs motion vs geometry) that fail in different regimes.
- Explicitly train the system to *allow disagreement* when charts are in different validity regimes.
- Evaluate coherence AUC against ground truth to verify that coherence correlates with correctness rather than merely being high.

In short: coherence must be earned by genuine independence. Otherwise the system becomes an agreement theater.

11.4 Stop rules: when discretization is necessary for action and language

Even if a percept filter can represent ambiguity honestly, downstream systems and human communication often require discrete commitments. You cannot always say "chart overlap continues." You must sometimes decide: grasp the object, pick the bounding box, name the person, draw the boundary.

So the system needs explicit **stop rules**: principled points at which it is allowed (or required) to collapse soft incidence and multiple hypotheses into a discrete summary.

Stop rules come in two families:

- **Action-driven stop rules**: a robot controller needs a hard grasp pose; an AR system needs a stable anchor; a navigation system needs a lane boundary. In these cases, discretization is required, but the CT/HoTT discipline demands that the decision be logged with its fragility: "I committed under boundary-zone uncertainty; here is the alternative."
- **Language-driven stop rules**: humans communicate with nouns and identities. If the tool is used in conversation or reporting, it must output something discrete. Here, the system can choose a best summary *plus* the audit object that supports it, so the discrete statement is paired with an inspectable witness.

A crucial point: stop rules should not be hard-coded as "always discretize." They should be conditional on coherence and conditioning. When the system is in a boundary zone, it may discretize only if forced—and then it should label the commitment as provisional.

This is where the paper's metaphysical humility becomes concrete: the system distinguishes "I must act" from "I know." It does not equate decision with truth.

11.5 Robustness stance: when to refuse transport and say "unknown"

Refusal is the safety valve of a coherence-first system. Without it, the system will overtransport, undersegment, and manufacture false witnesses. A robust CT/HoTT filter must treat "unknown" as a successful output in the right regimes.

We can define refusal conditions operationally:

- **No consistent witness exists**: transport residuals cannot be reduced without violating independent constraints (e.g., motion and landmark charts disagree beyond tolerances).
- **Boundary zone persists too long**: the system remains in overlap without stabilizing, suggesting insufficient data or nonstationary edits.
- **Constraint independence failure**: routes appear to agree but coherence AUC indicates poor calibration; agreement is untrustworthy.
- **Adversarial suspicion**: patterns known to induce overtransport are present (lookalikes, repeated textures, suspicious edit artifacts), prompting stricter witness requirements.
- **Task stakes exceed confidence**: if the application has high consequence (forensics, medical decision support), the refusal threshold is lowered, and the system outputs "cannot certify" more readily.

Refusal should not be silent. A CT/HoTT percept filter refuses with structure: it shows which constraint loops failed, where chart conditioning broke, and what additional data would be needed to resolve uncertainty (more frames, different angle, better lighting). This keeps refusal from feeling like a crash and turns it into guidance.

Finally, robustness requires a cultural stance: the system must be deployed with the understanding that it is a witness tool, not a truth oracle. The more beautiful the coherence renderings become, the greater the temptation to treat them as authority. The system's design must resist that temptation: by showing strain alongside confidence, by making boundary zones ordinary, by logging commitments, and by elevating "unknown" from embarrassment to integrity.

So the tradeoffs are real. A CT/HoTT filter can reduce snaps, but it introduces new failure modes: overidentification, undersegmentation, and coherence laundering. It also imposes compute and cognitive costs. The response is not to retreat from structure, but to govern it: diversity of routes, explicit stop rules, calibrated refusal, and UI discipline that reveals structure when it matters and stays quiet when it does not.

12. Ethics and Governance

A CT/HoTT percept filter is not just a new vision algorithm. It is a new *epistemic interface*. It turns hidden internal commitments into visible witnesses, and it can make identity continuity feel unusually "solid" because it is supported by transport traces and coherence overlays. That can be a moral improvement—less snap, more honesty—but it can also become a new kind of power: enhanced recognition, enhanced tracking, and enhanced persuasion. Ethics therefore cannot be an afterthought. Governance must be designed into the output contract itself.

The guiding principle of this section is simple: **the more coherent and inspectable the system looks, the easier it is to mistake it for an oracle.** We must counterbalance that affordance with constraints, contestability, and refusal modes.

12.1 Enhanced recognition and privacy risk

Any system that stabilizes identity under occlusion, lighting changes, and viewpoint sweeps is a privacy amplifier. Even if the system is "more honest," it can still enable harms: persistent tracking, re-identification, and surveillance creep.

The CT/HoTT filter raises privacy stakes in two specific ways:

- **Continuity as a product**: transport witnesses make identity tracking more robust over time and across transformations. That directly increases the capability to follow people in public spaces and to correlate appearances across contexts.
- **Audit trails as metadata**: the exported artifact (witnesses, strain traces, boundary zones) can become an additional fingerprint. Even if raw imagery is discarded, the structural traces may still allow re-identification or sensitive inference.

Therefore, a responsible deployment must treat the percept artifact as sensitive data. It is not a harmless overlay. It is a structured record of how identity was maintained.

Practical privacy safeguards follow from that premise:

- **Minimize retention**: default to ephemeral computation; do not store witness chains unless explicitly justified.
- **Scope reduction**: allow the system to operate in "non-identifying" modes (e.g., continuity for objects without personal identity binding, or continuity only within a short session).
- **Data separation**: separate raw imagery from percept artifacts; restrict their joint availability.

- **Differential access**: operators may be allowed to see stability indicators and boundary zones while being forbidden from seeing identity-level witnesses.
- **Explicit consent contexts**: in assistive and XR training contexts, the user's consent may justify stronger identity features; in public surveillance contexts, the default stance should be restrictive.

In short: better recognition is not automatically better society. The filter's capability must be bounded by purpose and permissions.

12.2 The temptation of "oracle sight": coherence is not truth

The system's most dangerous virtue is how convincing it can look. A commutative-looking diagram, a smooth transport ribbon, a high coherence score—these are psychologically persuasive. They can feel like "the system has seen the essence." But coherence is not truth. It is only internal consistency with a model class and a set of constraints.

There are at least four reasons coherence can be high while truth is wrong:

- the system's constraints omit the relevant causal factor,
- the routes are correlated (coherence laundering),
- adversarial inputs exploit the system's invariances,
- the world itself is ambiguous (multiple explanations fit).

Therefore, governance must include a deliberate de-mystification: the UI and the documentation must continually reinforce that *witnesses are evidence objects, not truth certificates*. The tool should display strain alongside confidence, and it should make refusal visible and respectable.

A concrete governance mechanism here is **claim-level throttling**:

- The system can output "coherence high" without outputting "identity confirmed."
- The system can output "transport plausible" without outputting "same person."
- Strong claims require stronger witnesses and more route independence.

This is not merely semantic. It changes how the tool is used. It encourages users to treat the system as an instrument that supports inquiry, not a machine that ends inquiry.

In the language of the paper: we must prevent "oracle sight." A CT/HoTT filter can be designed to resist oracle dynamics by making its own limitations legible and by providing institutional checks on where and how it can be applied.

12.3 Auditability, contestability, and calibrated uncertainty

The ethical promise of this project is not "better recognition." It is **better epistemics**: the system shows its work, exposes boundary zones, and logs forced discretizations. That promise becomes meaningful only if three properties hold in deployment:

- **Auditability**: outputs are inspectable; decisions are logged; witnesses can be examined after the fact; the system can replay why it made a commitment.
- **Contestability**: users can challenge outputs; alternative parses can be viewed; a reviewer can point to a specific constraint failure and say "this witness is weak here."
- **Calibrated uncertainty**: the system's confidence and abstention behavior match reality; high confidence correlates with correctness, and low confidence correlates with genuine ambiguity.

These properties should be treated as first-class acceptance criteria, not as nice-to-haves. A system that performs well but hides its flips is ethically worse in many high-stakes contexts than a system that performs slightly worse but is honest about uncertainty.

Contestability also implies *non-monopoly of interpretation*. The system should not bundle its own narrative as the only story. It should preserve alternative charts and show why it preferred one. That design choice prevents institutional capture where "the tool says so" becomes the end of discussion.

Finally, auditability must include **negative evidence**. When the system abstains, it should record what prevented witness construction. That makes abstention actionable and prevents it from being misused as an unchallengeable shrug.

12.4 Deployment constitution: logging, access limits, red-team gates, refusal modes

To operationalize ethics, we propose a deployment constitution: a set of non-negotiable governance requirements that travel with the system. The constitution is not a policy memo; it is a technical and procedural bundle.

Key clauses include:

- **Logging by default**: every hard commitment (identity bind, segmentation finalize, boundary event crossing that triggers a discrete change) must be logged with (i) the active charts, (ii) witness strength/strain, (iii) alternative hypotheses, and (iv) whether the commitment was forced by downstream action. Logs are the backbone of accountability.

- **Access limits**: restrict who can see what. “Inspect mode” should be gated. Identity-level witnesses should require higher privilege than stability indicators. In many deployments, only aggregate coherence metrics should be visible, not raw witness chains.
- **Purpose binding**: the system should be configured per use case with explicit claim limits. An accessibility tool may be allowed to maintain session-local continuity for the user; a public deployment may be forbidden from long-horizon re-identification. These limits should be enforceable technically, not merely promised socially.
- **Red-team gates**: before deployment, the system must be tested against overtransport and coherence laundering. This includes adversarial lookalikes, repeated textures, splices, deepfake manipulations, and correlated-route traps. Red-team evaluation should be repeated after updates, because model changes can alter calibration dramatically.
- **Refusal modes**: abstention must be available, visible, and protected from punishment. In high-stakes contexts, the system must be configured to prefer refusal over fragile commitments. Refusal should come with explanations (constraint cards) so that “unknown” is an accountable state, not an opaque failure.
- **Fail-closed behavior for identity**: when uncertainty rises sharply, the system should drop to weaker claims rather than doubling down. It is better to say “cannot certify” than to quietly smooth contradictions.

A constitution also implies *institutional roles*. Someone must own calibration monitoring, someone must review logs, and someone must be responsible for updating refusal thresholds when misuse is detected. Governance is not only a feature; it is a process.

The deeper ethical thesis is: exporting witnesses and coherence traces is only morally good if it strengthens humility and contestability. If it strengthens surveillance and authority, it becomes a weapon of persuasion. The same technical machinery can serve both. The difference is governance: how claims are throttled, how artifacts are retained, how access is gated, and how refusal is treated as integrity rather than weakness.

So the project’s ethical finish line is not a “better detector” or a “better recognizer.” It is a better covenant between system and user: the system shows structure, admits boundary zones, logs forced discretizations, and refuses when it cannot lift—so that seeing becomes a disciplined witness rather than an oracle.

13. Conclusion and Forward Work

This paper began with a simple perceptual stress test: the cube–octahedron duality and the stubborn difficulty of "seeing" a smooth transition in a property that feels discrete, like "3 faces meet here" versus "4 faces meet here." Ordinary vision—human and machine—tends to treat such refactorizations as ontological events. We experience a snap. The wager of this work is that the snap is not forced by geometry; it is induced by the perceptual primitive. If perception is object-first and discretizes early, then changes in the discrete census feel like reality breaks. If perception is relation-first and carries continuity as witnessed transport across charts, then the same discrete changes can be rendered as ordinary chart transitions.

This conclusion restates that thesis cleanly, specifies what would count as a real advance, and proposes next steps that are both technical and governance-aware.

13.1 Thesis restated: phase changes become chart changes under witnessed transport

The central thesis is not "polyhedra can morph smoothly." We already know they can, and we already know discrete summaries can change along smooth deformations. The thesis is about *how a perceptual system represents that fact*.

When a system treats discretizations (edges, segments, identities) as primitives, a refactorization is experienced as a phase change in the world. That is the snap. Under this regime, a change in vertex valence or a split/merge in segmentation is not merely a new description; it is perceived as "the thing changed into another thing."

A CT/HoTT-inspired filter proposes a different ordinary:

- A percept is not a single parse; it is a family of charts with domains of validity.
- Identity is not equality; it is transport along time and transformation, supported by explicit witnesses.
- Boundary events are not flip moments; they are overlap zones where chart stability trades hands.
- Coherence is not truth; it is agreement across routes, exported as a trace and paired with strain and refusal.

Under this discipline, what looks like a discrete phase change becomes a chart change: the underlying continuous substrate persists; the descriptive coordinate system refactors; the system shows where and why; and it refuses when the lift cannot be made.

That is the core metaphysical move, translated into engineering terms: phase change becomes chart change under witnessed transport.

13.2 What would count as a genuine advance: exported witnesses that users can inspect

The most important claim of the paper is not that we can build a robust model internally. Many systems already internalize sophisticated structure. The genuine advance would be *exporting that structure as an inspectable artifact*—turning perception into something like a readable audit trail.

Concretely, a real advance would mean:

- The system outputs multiple chart renderings (contours, regions, landmarks, motion) with local conditioning, not just one mask.
- The system outputs transport witnesses across time (and across perturbations) that can be replayed, not just track IDs.
- The system outputs witness objects relating competing parses—refinement/coarsening correspondences or explicit "no witness" flags—rather than silently choosing one and flipping later.
- The system marks boundary zones as zones and logs discrete commitments when forced by action.
- The system's coherence signals are calibrated: high coherence correlates with correctness, low coherence correlates with ambiguity, and refusal is a trained success mode, not a failure.

Most importantly, *users must be able to inspect and contest the witnesses*. If the system claims "same person," it should show the transport chain and where it is strained. If the system flags a synthetic edit, it should show the constraint inconsistency rather than a verdict. If the system changes its mind, it should show the overlap zone and the reason for chart handoff.

If we cannot deliver inspectable witnesses and contestable traces, then the project collapses back into a hidden-trick model with new vocabulary. The exported artifact is the point.

13.3 Next steps: prototype filters, benchmark releases, and responsible rollout

Forward work divides naturally into three tracks: prototypes, benchmarks, and governance.

Prototype filters
A near-term goal is to build two reference implementations:

- **Route A prototype (algorithmic)**: a field-to-charts-to-incidence pipeline that outputs chart overlays, coherence heatmaps, boundary zones, and simple transport ribbons for video. The purpose is clarity and auditability, not SOTA performance. This prototype can also generate weak supervision for Route B.

- **Route B prototype (learned)**: a multi-chart decoder model with explicit transport objectives and witness prediction, trained with synthetic sweeps and controlled occlusions. The key deliverable is not accuracy alone but an exported percept artifact with witnesses and refusal.

In both prototypes, the UI is not optional. The percept rendering is part of the system. A prototype that cannot render its own charts and witnesses has not implemented the paper's thesis.

Benchmark releases
The second track is to release benchmarks that measure snap behavior and epistemic honesty:

- invariance profile sweeps (occlusion, lighting, viewpoint),
- ambiguity sequences that reward non-collapse and continuous reweighting,
- synthetic media tests focused on constraint-consistency,
- and the **polyhedron-movie task** as a clean calibrator for graded incidence and chart overlap.

Metrics should include time-to-stability, reversal geometry, coherence AUC, calibrated abstention, and snap explainability (changes accompanied by boundary-zone logs).

This benchmark track is crucial because it makes the thesis falsifiable. Without it, the project risks becoming a compelling aesthetic without measurable behavioral improvement.

Responsible rollout
Finally, governance must ship with the system:

- default logging of hard commitments,
- access controls that gate identity-level witnesses,
- refusal modes that are protected and calibrated for stakes,
- red-team gates targeting overtransport and coherence laundering,
- and purpose-bound configurations that prevent silent escalation into surveillance tooling.

A responsible rollout also implies deployment discipline: start in low-stakes contexts (creative tooling, XR training, scientific imaging where analysts already expect uncertainty), learn failure modes, and only then consider higher-stakes domains with strict claim throttles.

If these next steps are pursued, the result would be something rare: not just a better classifier, but a new *genre of perception output*—a structured rendering that turns refactorization from a snap into an explainable transition, and that invites the user into the act of seeing as an audited process rather than a verdict.

That is the forward work: build the percept artifact, test it against snaps, and deploy it under a constitution that keeps coherence from becoming an oracle.

References

Foundations: Category Theory, Applied Category Theory, and HoTT

Mac Lane, S. *Categories for the Working Mathematician.* 2nd ed., Springer, 1998. Springer

Leinster, T. *Basic Category Theory.* Cambridge University Press, 2014. Cambridge University Press & Assessment+1

Fong, B., and Spivak, D. I. *Seven Sketches in Compositionality: An Invitation to Applied Category Theory.* arXiv:1803.05316, 2018. arXiv+1

The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics* ("The HoTT Book"). Institute for Advanced Study, 2013. Homotopy Type Theory+1

Percept-as-witness motivation inside this project

Youvan, D. C. "Duality as an Incidence Category Picture: A Categorical Rendering of the Cube–Octahedron Pair." Preprint/manuscript, Dec 28, 2025.

Core CV backbones and "continuous field" representations

He, K., Zhang, X., Ren, S., and Sun, J. "Deep Residual Learning for Image Recognition." arXiv:1512.03385, 2015. arXiv

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis." arXiv:2003.08934, 2020. arXiv

Park, J. J., Florence, P., Straub, J., Newcombe, R., and Lovegrove, S. "DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation." CVPR, 2019. CVF Open Access

Uncertainty, calibration, abstention, and "refuse transport" mechanics

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. "On Calibration of Modern Neural Networks." arXiv:1706.04599, 2017. arXiv+1

Lakshminarayanan, B., Pritzel, A., and Blundell, C. "Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles." arXiv:1612.01474, 2016. arXiv+1

Hendrycks, D., and Gimpel, K. "A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks." arXiv:1610.02136, 2016. arXiv+1

Gal, Y., and Ghahramani, Z. "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning." arXiv:1506.02142, 2015. arXiv+1

Kendall, A., and Gal, Y. "What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?" arXiv:1703.04977, 2017. arXiv+1

Geifman, Y., and El-Yaniv, R. "Selective Classification for Deep Neural Networks." NeurIPS, 2017. NeurIPS Papers

Equivariance and higher-order structures beyond pairwise graphs

Cohen, T. S., and Welling, M. “Group Equivariant Convolutional Networks.” arXiv:1602.07576, 2016. arXiv+1

Ebli, S., Defferrard, M., and Spreemann, G. “Simplicial Neural Networks.” arXiv:2010.03633, 2020. arXiv+1

Additional “Route B” staples (IDs suitable for bibliography completion pass)

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. “A Simple Framework for Contrastive Learning of Visual Representations (SimCLR).” arXiv:2002.05709, 2020.

Grill, J.-B., Strub, F., Altche, F., et al. “Bootstrap Your Own Latent (BYOL).” arXiv:2006.07733, 2020.

Caron, M., Touvron, H., Misra, I., et al. “Emerging Properties in Self-Supervised Vision Transformers (DINO).” arXiv:2104.14294, 2021.

He, K., Gkioxari, G., Dollar, P., and Girshick, R. “Mask R-CNN.” arXiv:1703.06870, 2017.

Teed, Z., and Deng, J. “RAFT: Recurrent All-Pairs Field Transforms for Optical Flow.” arXiv:2003.12039, 2020.

Polyhedra and duality background for the “polyhedron-movie” benchmark

Coxeter, H. S. M. *Regular Polytopes.* 3rd ed., Dover, 1973.

Ziegler, G. M. *Lectures on Polytopes.* Springer, 1995.

Grunbaum, B. *Convex Polytopes.* 2nd ed., Springer, 2003.

Witnessed Transport
Over Chart Changes
Category-Theoretic Perception